THE UNIVERSITY OF NOTTINGHAM
Dept of Electrical and Electronic Engineering

EDGE DETECTION USING NEURAL NETWORK ARBITRATION

by

Mário António da Silva Neves Ramalho
Licenciado em Engenharia Mecânica
Mestrado em Engenharia Mecânica

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

May 1996

Acknowledgements

First of all, to Dr Kenneth Mervyn Curtis for his invaluable advice and help and friendship during the course of this PhD. To all members of the Parallel Processing Specialist Group. Among them Merv, Ahmed, Zaha, Shaun and James deserve a special Thanks.

To the Junta Nacional de Investigação Cientifica e Tecnológica, which, from the Programs Ciência and Praxis XXI which provided financial support for my studies. To GCAR, IST, UTL. Among them a special thanks to Dr Caldas Pinto and Eng. João Sousa.

To my wife, Maria Celeste, to my family and friends who helped me through these years. (Specially to the 'Grupo do Bacalhau' and 'Goodfellas', and all the others)

Finally to all who helped me during the preparation of this thesis, and who offered images from their work and put them at my disposal to apply my research, even if not used (Paul Crump , Pilar Sepulveda and Dr Hai Bo. Diva Ferreira).

To:

Joana Ramalho and Joaquim Ramalho

wherever they are

Mario Pedro and Maria Ines

Whenever they arrive

but above all

to

Maria Celeste da Fonte Corte

for the time she waited.

This work was supported by the grants BD 1719/91 from Programa Ciência and 3950/94 from Praxis XXI, both from the Junta Nacional de Investigação Cientifica e Tecnologica

ABSTRACT

A human observer is able to recognise and describe most parts of an object by its contour, if this is properly traced and reflects the shape of the object itself. With a machine vision system this recognition task has been approached using a similar technique. This prompted the development of many diverse edge detection algorithms.

The work described in this thesis is based on the visual observation that edge maps produced by different algorithms, as the image degrades, display different properties of the original image. Our proposed objective is to try and improve the edge map through the arbitration between edge maps produced by diverse (in nature, approach and performance) edge detection algorithms. As image processing tools are repetitively applied to similar images we believe the objective can be achieved by a learning process based on sample images.

It is shown that such an approach is feasible, using an artificial neural network to perform the arbitration. This is taught from sets extracted from sample images. The arbitration system is implemented upon a parallel processing platform. The performance of the system is presented through examples of diverse types of image. Comparisons with a neural network edge detector (also developed within this thesis) and conventional edge detectors show that the proposed system presents significant advantages.

1 INTRODUCTION

"The real question is not whether machines think but whether men do"
(B.F.Skinney, Contingencies of Reinforcement)

1.1 Preamble

The aim of this chapter is to introduce general issues concerning edge detection and to provide a brief overview of the need for and the description of the work to be presented within this thesis. The following section presents an introduction to the problem of image processing, an analysis of edge detection and it's importance. Section 1.3 explains the importance of the research in edge detection and what is new and novel about the approach presented. Next. in section 1.4 the objective and tasks of the work are described. Finally the layout of the remainder of this thesis is presented.

1.2 Digital Image Processing and Computer Vision

Digital image processing is an area that has seen great development in recent years. It is an area which is supported by a broad range of disciplines. where signal processing and software engineering are among the most important. Digital image processing applications tend to substitute or complement an increasing range of activities. Applications such as automatic visual inspection. optical character recognition. object identification . etc.. are increasingly common.

Digital image processing studies the processing of digital images, i.e., images that have been converted into a digital format and consist of a matrix of points representing the intensities of some function at that point. The main objectives are related to the image improvement for human understanding and 'the processing of scene data for autonomous machine perception' [48].

This later task normally comprises a number of steps. The initial processing step is the segmentation of the image into meaningful regions, in order to distinguish and separate various components. From this division, objects can then be identified by their shape or from other features. This task usually starts with the detection of the limits of the objects, commonly designated as edges. Effectively a human observer is able to recognise and describe most parts of an object by its contour. if this is properly traced and reflects the shape of the object itself. In a machine vision system this recognition task has been approached using a similar technique, with the difference being that a computer system does not have other information than that which is built into the program. The success of such a recognition method is directly related to the quality of the marked edges.

Under 'engineered' conditions. e.g. backlighting, edges are easily and correctly marked. However, under normal conditions where high contrast and sharp image are not achievable. detecting edges become difficult. Effectively. as contrast decreases the difficulty of marking borders increases. This is also the case when the amount of noise present in the image increase. which is 'endemic' in some applications such as x-rays. Common images. e.g. from interior scenes. although containing only small amounts of noise. present uneven illumination conditions. This diminishes contrast which affects the relative intensity of edges and thus complicates their classification. Finally.

image blur due to imperfections in focus and lens manufacture smoothes the discontinuities that are typical from edges and thus once again makes the edges difficult to detect.

These problems prompted the development of edge detection algorithms which, to a certain degree of success, are able to cope with the above adverse conditions. Under suitable conditions most of the edge detection algorithms produce clear and well defined edge maps, from which objects within the image are easily identified. However, the produced edge maps degrade as the image conditions degrade.

The work described in this thesis is based on the visual observation that edge maps produced by different algorithms, as the image degrades, display different properties of the original image. Not only misplacements of the shape occur, spurious features appear and edge widths differ from algorithm to algorithm. It may be hypothesised that edge maps produced by different algorithms complement each other. Thus it may be possible to override some of the vagueness by comparison between different edge maps.

Our proposal is to try and improve the edge map through the arbitration between edge maps produced by diverse (in nature, approach and performance) , edge detection algorithms.

1.3 Objectives

Our main objective, is to investigate the feasibility of an arbitration system for generalised edge detection in images. Similar approaches where the information from diverse methods is fused into one map have not been

reported before. The development will be carried out in successive steps, starting with the selection of edge detection algorithms that form a suitable basis to test the feasibility of such an approach. This will be followed by the development of the system and its testing on various images. This involves several stages in defining images and in the selection of suitable images to test the hypothesis.

Neural networks are well known for their ability to arbitrate between different inputs. Another objective of the research is to investigate neural networks for generalised edge detection so that the advantages or disadvantages of the arbitration system can be assessed.

The development of the arbitration strategy is another goal of the research. The definition of suitable edge detector tuples, the strategy to implement them, and as neural networks are used, the research of suitable learning sets are stages that are also considered.

Finally, a comparison criteria that allows for the assessment of the developed technique, in comparison to others, is also needs defined.

1.4 Outlook

The structure of this thesis is as follows:

Chapter Two starts with a discussion on the concept of an edge. Effectively this concept of an 'edge' lacks an objective definition. In the edge detection literature several different edge concepts are described, sometimes even mixed in the same reference. Some apparent failures of edge detection

algorithms are due more to a misunderstanding of the concept of an edge rather than to a failure of the algorithms themselves. There then follows a literature review of edge detection algorithms and a comparisons between their performance drawn. Finally, a literature review on neural networks is presented.

Chapter Three presents the mathematical background used for the development of the analysis technique described within this thesis. It presents the mathematical theorems that prove that such an approach is feasible.

Chapter Four describes the implementation of the core system upon a parallel processing platform so as to maintain computational competitiveness in practical applications. A review of basic concepts in parallel processing is followed by the description of the test performed in search of the most efficient paradigm.

Chapter Five consists of two parts. In the first, examples of edge detection techniques and their performance are presented. The selection of suitable images upon which to carry out the tests is performed and the different edge detection operators performance on such images analysed. The second part describes the development of a neural network for edge detection. Examples of the performance of the solutions are presented along with the tests performed during the development of the neural network edge detector.

Chapter Six presents the arbitration strategy and the development of a system to perform it. Several examples are presented. and comparisons between the arbitrated edge maps and the techniques used to obtain the edge maps are

carried out. Finally, the solutions are compared to the neural network edge detector, and the advantages of the approach being researched presented.

Finally, in Chapter Seven conclusions are drawn. The advantages and disadvantages of the technique being researched are described. Proposals for further work suggested by the researched techniques are also presented.

2 EDGE DETECTION REVIEW

A complete survey of edge detectors is not a simple task, and can even be confusing. (I. Abdou)

2.1 Introduction

Edge detection is a very common topic within the literature on image processing . Although it is not a recent research field it still originates several papers every year. The number of approaches to the problem is great, covering almost all mathematical fields known. However, the complexity of the problem and its ill defined nature means that there are still many problems associated with the field. Through this chapter we will review the different approaches. However, before such an extensive work is carried out, basic image processing concepts will be analysed.

2.2 Image Processing

Image processing dates back to the beginning of the century. when the problems that arose due to the transmission of an image down a cable where reported upon [48]. In more recent years. digital computers have allowed for an increase in the number of applications of image processing. Image processing applications are common in medicine. biological sciences. archaeology, surveying. manufacturing processes, robotics. etc. Vision. which

is one of our senses, is probably the sense which provides us with more information then any other about the outside world.

When analysing images it should be kept in mind which part of the information is within the image, and which part is overlaid from our perception system. Indeed, in some situations we perceive details that are extrapolated from our knowledge about the objects within the scene, and thus, added from information that are external to the image. Thus it is very difficult to quantify some features within the image.

2.3 Edge detection objectives

The interest in digital image processing came from two principal application areas: improvement of pictorial information and processing of scene data for autonomous robot classification within autonomous machine perception. In the second area. where the most primordial motivation of this thesis is based and in which edge detection is used. the first processing steps are the identification of meaningful areas within the picture. This process is called segmentation. It represents an important early stage in image analysis and image identification. Segmentation is a grouping process in which the components of a group are similar in regard to some feature or set of features. Given a definition of "uniformity", segmentation is a partitioning of the picture into connected subsets. each of which is uniform. but such that no union of adjacent subsets is uniform [129]. There are two complementary approaches to the problem of segmenting images and isolating objects - boundary detection and region growing. Edge oriented methods generally lead to incomplete segmentation. the resulting contours may have gaps or there may be additional erroneous edge elements within the area. The results can be converted into complete image segmentation by suitable post processing methods. such as contour

following and edge elimination. Region growing is a process that starts from some suitable initial pixels and using iterations neighbouring pixels with similar properties are assigned, step by step, to sub regions. A suitable basis for this type of segmentation could be a thresholding process.

Marr´s Theory of Vision [106][108] starts from the construction of the 'raw primal sketch' in which changes in intensity are made explicit and from where edges are then extracted. Also Artificial Intelligence approaches to vision used a frame description of objects, from which and through the use of reasoning systems objects could be identified. There are other systems where edge detection is used, as a basic tool, in visual inspection systems for quality control.

2.4 What is an edge?

There is no universal definition of an edge. As the main purpose of defining edges is segmentation or measurement of the objects within an image, edges should represent the limits of the objects within the image. An edge is generally stipulated as the result of any sharp alteration in the characteristics of the image.

The intensity of reflected light is a function of the angle of the surface to the direction of the incident light. This leeds to a smooth variation of the light intensity reflected across the surface as its orientation to the direction of the light source changes, which can not be considered as an edge. Also shadows give sharp changes in the brightness within an image of a smooth and flat surface. This does not represent the limit of an object. In the other extreme, such as in the case of technical drawing, where there are thin lines drawn, where no discontinuity on the represented object exists, but which are

important for the understanding of the shape of an object. The relation between edges and grey level discontinuities is not clear, and a decision can only be made where an understanding of the image exists (which, in some way, is the ultimate goal of the whole image processing process). As Vicky Bruce [14] states

> "Clearly, there is a relationship between the places in an image where light intensity and spectral composition change, and the places in the surroundings where one surface or object ends and another begins, but this relation is by no means a simple one. There are a number of reasons why we can not assume that every intensity or spectral change in an image specifies the edge of an object or surface in the world" (Bruce, op.cit, page 91)

Several definitions of edges have been proposed by different authors. Italics highlight the vagueness that the authors included in their proposed definitions. Sonka et al [156] proposed it as

> "... pixels where this function (brightness) changes *abruptly*" (op cit., page 76)

Pratt [137]

> "Local discontinuities in image luminance from one level to another are called luminance edges. (...) The definition of a luminance edge is limited to image amplitude discontinuities between *reasonably* smooth regions" (op. cit., page 491)

Rosenfeld and Kak [145]

> "An edge: the grey level is *relatively consistent* in each of two adjacent, extensive regions, and changes *abruptly* as the border between the regions is crossed" (op. cit., page 84)

Gonzalez [49]

> "...an edge as the boundary between two regions with *relatively distinct* grey-level properties" (op.cit, page 334)

From the above definitions the words printed in italic should be retained as none of the authors give a precise meaning to each of these words. As Blicher stated:

> "Everyone agrees that a perfect step function should give an edge, but there has been no adequate criterion put forth to classify any other function as edge, no-edge" (Op cit., page 4)

Perhaps the most adequate definition was that given by A. Blicher (Op. cit.):

> "Edges, it seems, are a lot like obscenity, for as Mr Justice Potter Stewart wrote of obscenity [Jacobelis vs. Ohio 1964], he may not be able to define it, 'But I know it when I see it'. (Op. cit., page 4)

Effectively there are many well known paradoxes in which an edge or contour is clearly seen where none physically exists. This is due to the characteristics of our perception capabilities, and our tendency to group similar information. as described in Gestalt's approaches to perception [98] or to infer edges from the context [87] of the image.

Kak's and Pratt's definitions include fine and low contrasted textures or small grey level gradients in the image. There are many examples of such types of textures such as newspaper images and the images presented within this thesis as the printer used. although of a high resolution (600 dpi). still prints dithered images. instead of grey level images. However. a clear distinction between

texture and grey level areas are not presented by any of the above cited authors.

The alternative is to accept that every discontinuity in the grey level of an image represents an edge. However, even this definition can not be accepted without restrictions, as for instance noise present in the image produces local discontinuities in the grey level of images which can not be accepted as edges. If such a definition is accepted without restrictions none of the algorithms presented mark false edges or wrong edges. They just mark their own definition of discontinuity, and thus their own definition of an edge.

The weakness and limitations of the concept can be shown through an image with two areas. Between them grey levels present a linear varying discontinuity from 0 to δ. Lets assume that the two areas present a linear varying grey level with the ranges [0 .. n] and [0 ..n+δ] respectively, and such that $n/x << \delta$. A schematic three-dimensional graph of such an image is presented in figure 1.

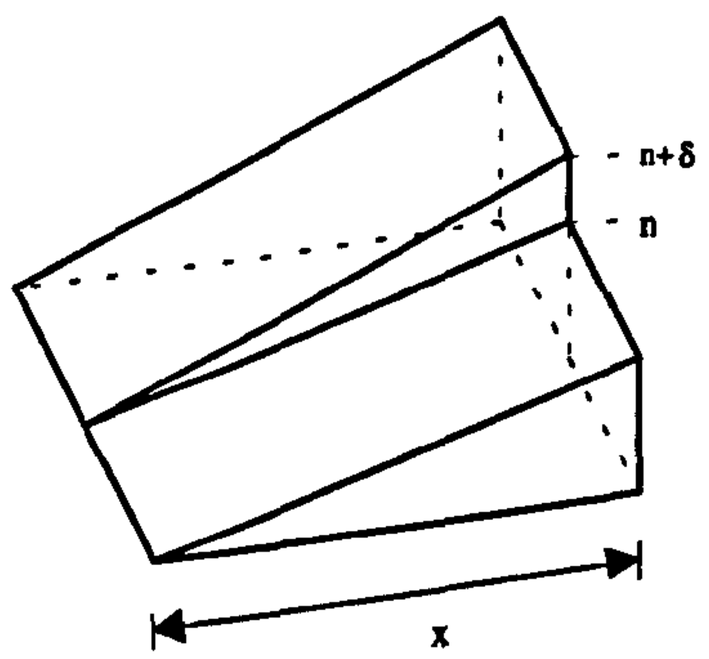

Figure 1: Grey level 3D plot of an image which consists of two distinct areas

In an image defined like this the edge, as the discontinuity between the two surfaces, will be marked by the same operators with a different extension

depending of the value of the parameter *n*, although the discontinuity itself is independent from it.

All the above definitions include some ambiguity in the definition of an edge. Effectively it will be only known at the end of processing, when segmentation has been performed, where the edges are. An edge is a subjective entity, as defined by Blicher. As this author said "The term 'edge' has been fairly abused and we will continue that tradition here". (Blicher, op.cit)

Edges appear with several profiles, where the most common are step edges. However ramp edges and roof edges (figure 2) are also common.

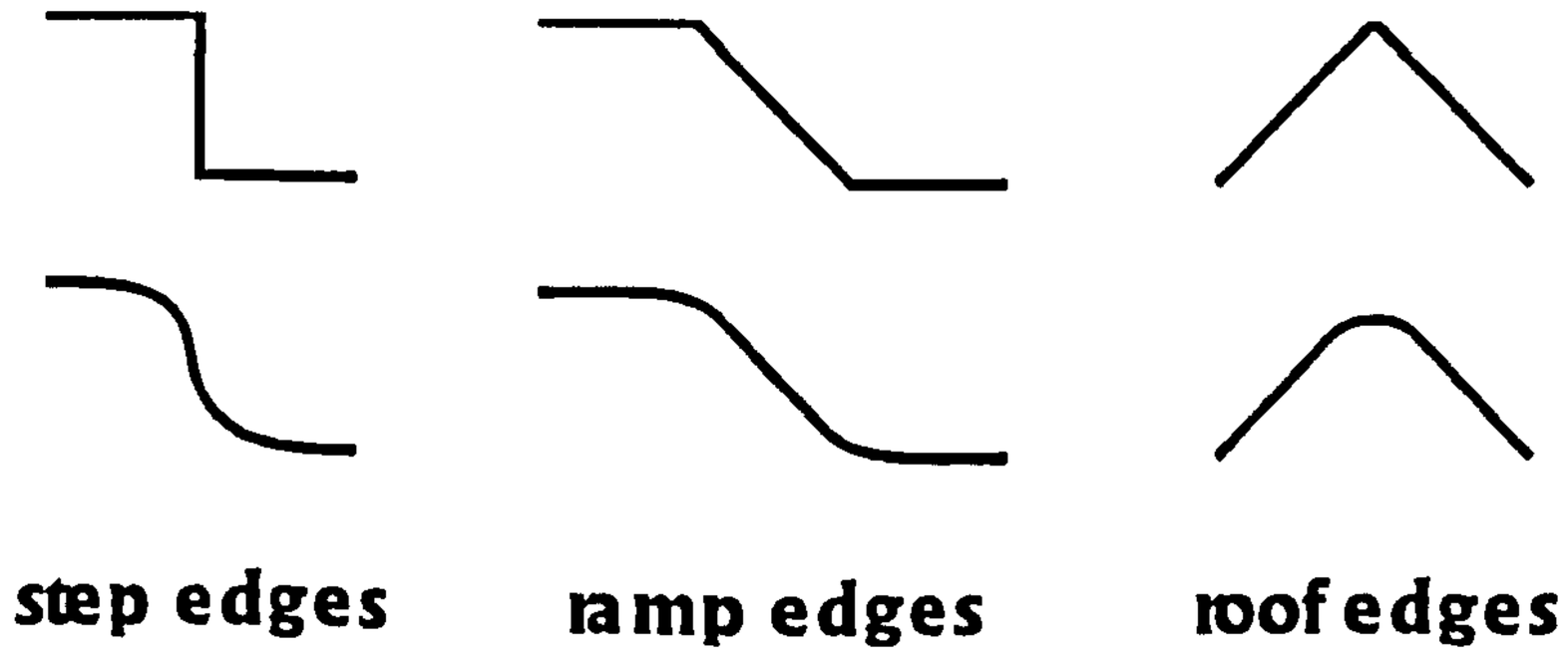

Figure 2: Common models for edge profiles

Edges also appear with several shapes, like curves or straight lines. The most common shape in our environment is vertical or horizontal straight lines (although this can not be transposed to image processing as a correct perspective will be very difficult to obtain). Some particular scenes (for instance blood cells) do not have straight edges at all. So a particular edge shape can not be assumed a priori. All these shapes have associated with them some quantity of noise, inherent to the acquisition process. These problems can be seen in figure 4, which is a three dimensional plot of the image from

the C key of the keyboard image in figure 3.

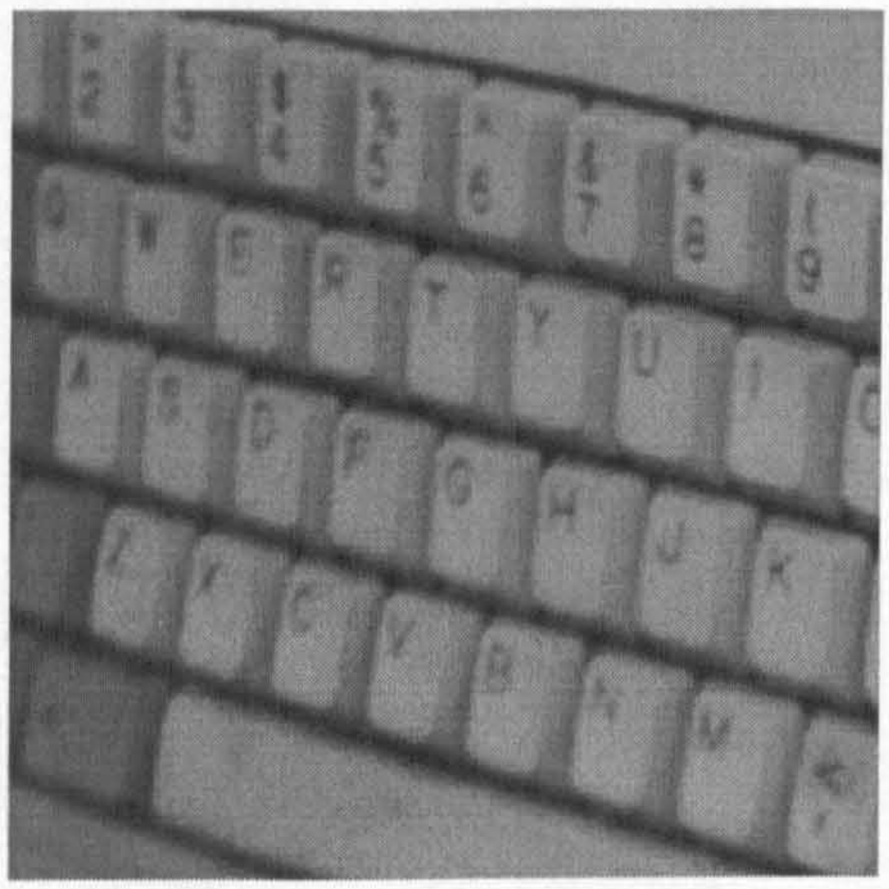

Figure 3:Image of a keyboard

In particular the grey level variation in the C print (clearly visible in the middle as a re-entrance, which is darker than the key) is not abrupt.

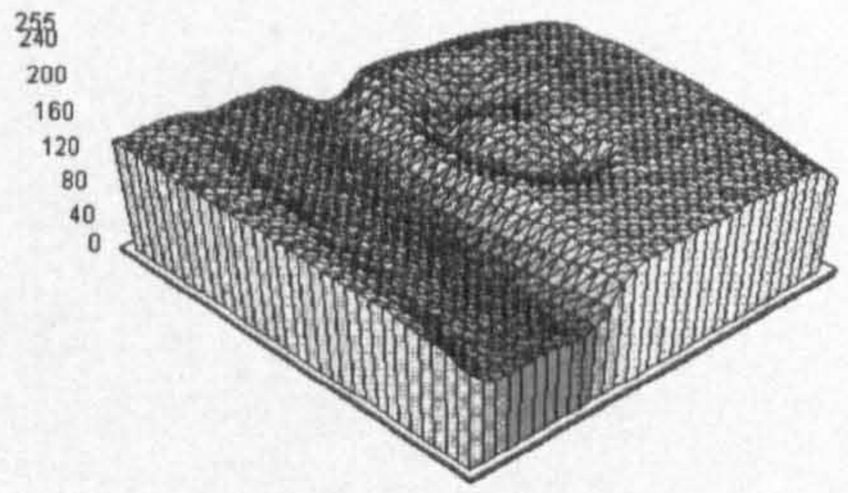

Figure 4: Three dimensional plot of the image from the C key of the computer keyboard

With the known exception of some images used in robotics (which are acquired with very high contrast, upon which binary thresholding is carried out), all images from real scenes have a small amount of noise. This can also

be originated from external sources to the acquisition system, e.g. electromagnetic interference.

2.5 What is Edge detection?

Edge detection is the process of extracting the edges from a digitised picture . It is generally assumed as the process of extracting the grey level discontinuities of an image, independent of their origin.

Several techniques exist, such as sharpening or crispening, which are designed to increase the visibility of general low contrast edges. These often lead to an increased perception of detail. Such techniques were not used in the work reported in this thesis.

2.6 What is an edge detector?

The definition of an edge detector is not generally accepted. Indeed Sobel masks can be classified as an edge detector or an edge enhancement filter. The main difference arises from the requirements for the edge detection output, whether it be a decision or, a likelihood measure. We will assume a wider definition of 'edge detector' based on two pictures. The first, designated as image A, is the picture of the scene being analysed. The second, designated as image B, consists of points or lines, marking the localisation of the edges of the first image. An edge detector, is the operator that allows us to pass or transform image A into image B. Following this definition and using the initial example, the Sobel masks, albeit the core, are part of the operator that allows such a transformation. Thus, an edge detector can be the result from the composition, of several filters. As an example, an edge detector can be composed of an edge enhancement operator and a decision operator. In general

the process of extracting the edges from an image usually follows three steps. First noise is removed using a low pass filter. This step is sometimes imbibed in the edge enhancement or measurement of the discontinuity. The next step is to analyse (or convolute) the image with some operator that gives as its output some measure of the "edginess" of the pixels or neighbouring pixels. Finally from this assessment it must be decided which edges are relevant, or what pixels are accepted as part of an edge. This can be done by simply thresholding the output of the operator (which means that an edge is a discontinuity greater then something, and that certain edges will vanish), using a local threshold of the output as its average value could vary from place to place. Finally, an algorithm is applied that links the edges or limits the marked edges to some particular kind of shape. Some authors add line thinning , which is assumed as a post processing stage. Noise removal is considered a pre processing stage unless implicitly included in the edge enhancement operator.

2.7 Solicited characteristics of an edge detector

An edge detector (or the edge enhancement operator) should present characteristics that allow for the correct interpretation of an edge and preferably alleviate the task performed by any subsequent modules.

Amplitude response variance to edge orientation - Preferably an edge detection scheme must be isotropic, and have an invariant response with edge orientation. It facilitates thresholding definition, otherwise a simple threshold gives origin to broken edges, where they do not exist. If the threshold is set too low, a width variation in the detected edge appears. This is related with the following characteristic.

Rapidly declining edge gradient response as the detector mask moves away from the centre of the edge. This factor is associated with precision and localisation.

Resolution - Responses due to near edges should be clearly distinct and should not interfere with one another. However, the notion of near edges is subjective to the user.

Good localisation - Marked edges should be in the true position, This particular property could be fundamental in some applications. For a segmentation process it may be better to have all the edges with a small bias than a few edges in the correct place.

Robustness or noise insensibility - Some images are inherently noisy (e.g. X-rays). This is due to the characteristics of the objects being analysed, which impose the use of low intensity radiation [43]. All images obtained from sensors have, at least, a small amount of noise. The image acquisition process could also have noise superimposed from several other sources (electromagnetic interference, for instance). An edge detector process should be insensitive to the noise or at least not alter its performance with small variations of the noise level.

Robustness to Blurring -It is virtually impossible to guarantee the absolute focusing of the image (due to optical effects). The edges that appear in an image from a real scene are not perfect and do not follow exactly the mathematical model used. An edge detector must have a range of acceptance of data with which it can deal. The problem of blurring is also included in the edge definitions presented and leaves open to the user the extent to which an edge is still an edge.

Sensitivity - The amplitude of an edge could vary with illumination characteristics. Some of the edges in a real scene can be lost, due to several factors, such as poor contrast. A good edge detector must mark the edges independently of the contrast between the objects. However, the presence of noise and the need for a threshold can limit this.

Computational load - The time for running an algorithm could determine its practical use. This factor is application, hardware and software dependent. Also total storage requirements should be kept to the minimum.

It can be said that the most important characteristics of an edge detector are *accuracy, sensibility, precision and isotropy.*

However these characteristics could be conflicting and some concessions may have to be made.

2.8 Important edge map features

The objective of edge detection is to detect edges. So the main objectives should be to evaluate the results rather than the operators themselves. Important characteristics of an edge are:

Shape - The line obtained should reflect as accurately as possible the shape of the phenomena that originates it.

Position - The detected edge should be in the correct position.

Thickness - The detected edge should be as thin as possible (ideally 1 pixel wide), for localisation and following algorithms.

Coherence - The detected edge should be a continuous and preferably uniform line, to facilitate analysis.

According to Hall [1] :

> "Edge images created for scene matching must be capable of meeting the following requirements: (i) retain salient edges of the objects to be matched, (ii) be relatively free of spurious edges extracted from the background and (iii) tolerate minor geometric misregistration" (op.cit. page 505).

In general it is difficult to define which are the salient edges within a scene and they must be decided upon based on the application goals.

2.9 A review of edge detection techniques

2.9.1 Introduction

Edges are assumed as discontinuities in the grey level of the image. The easiest way to discover these discontinuities is to shift the image in two perpendicular directions and subtract the resulting images from the original. Each of the difference images will have a value zero at all points which have the same grey level as the neighbour in the shift direction and the difference value otherwise. If noise is present in the image, all the image will have a non zero value. We must now define a threshold between the values resulting from the noise and the values resulting from the discontinuities. This process is clear if the range of the values are not overlapping, but this is not the general case. This technique, empirical as stated here, will be referred to later in a more formal way (it is equivalent to the Roberts mask). It is referred to here because it contains the main parts of the algorithms and could reveal the main weakness of the different algorithms. These are noise sensitivity and a limit to the smallest detectable discontinuity imposed by the noise present.

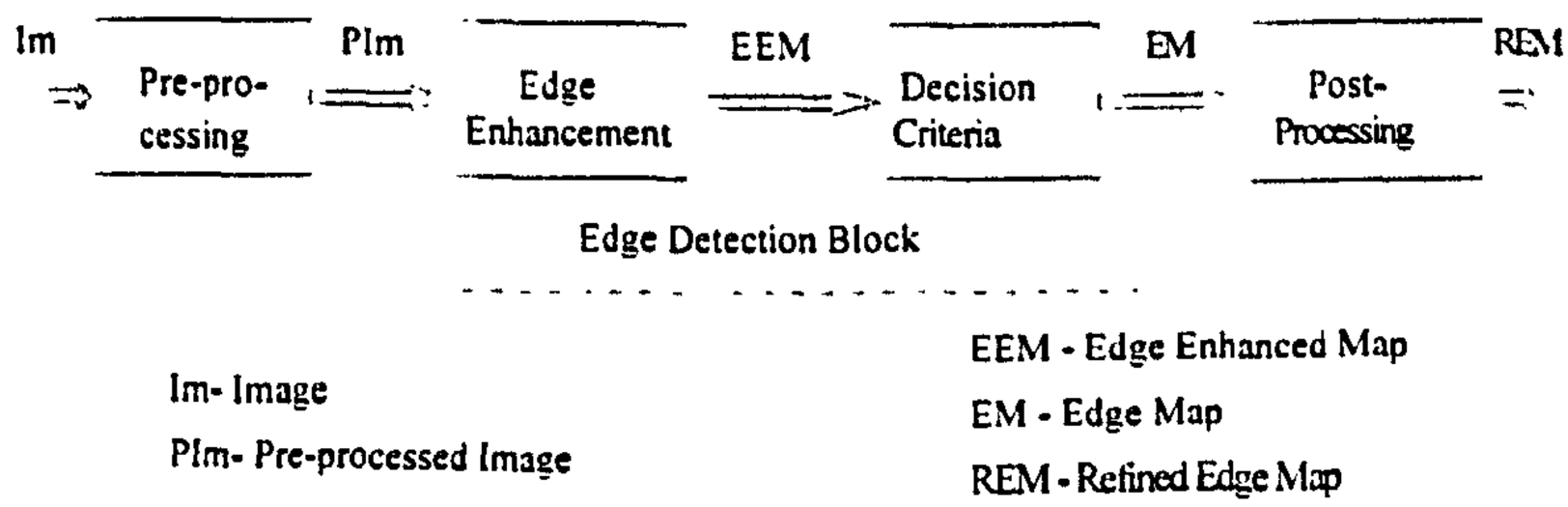

Figure 5: Edge detection scheme

Figure 5 shows a generalised edge detection scheme. An edge detection algorithm usually contains two main parts, performed sequentially: an edge enhancement filter and a decision process. In the first part, a filter is applied to the picture to enhance the discontinuities of the image (which is also called an edge detector by some authors). This picture will be called an edge enhanced picture. To this intermediate image, containing the candidate points, some decision criteria is applied to select the correct points. At this stage we will obtain an image containing only the edges (which will be called an 'edge map').

This two part scheme is not always explicit. In a large number of the algorithms the two parts can be combined producing different properties. Thus different approaches to edge enhancement will be described separately to the different approaches for decision making.

To these two main steps could be added a previous smoothing or enhancement step [187]. The smoothing step has the function of noise filtering or regularization of the solution. Although it is usually carried out using a low pass filter. this smoothing step could be implemented using different methods (anisotropic diffusion. for instance). The smoothing step is in some cases absorbed by the filter or mask itself [17][107]. Indeed some common masks can be seen as a composition of smoothing and enhancement masks [59]. The existence of this step is criticised by some authors [57] . and considered an important pre-processing step by others [170] . due to the resulting blurring of the edges caused by the smoothing filter

A post-processing step is added by some authors, to improve the edge enhanced map. This is needed due to gaps in the edge map or artificial features created by the algorithms (usually due to the noise present). This is normally achieved through thinning the edges, testing their continuity and hence filling existing gaps, removing created edges etc. [181] [92] [154].

It is difficult to associate edge detection techniques due to the multiple mathematical supports used and the different approaches in which a particular technique can be included. Effectively is not only different edge models that are assumed by the different techniques, as sometimes a different edge concept is subjacent to the method proposed. Several criteria could be used to group different techniques together. For instance, the Roberts operator could be considered an empirical technique, a derivative technique (as stated by Roberts [140]), or as a Hueckel type operator (using the sum of the absolute values. as stated by Rosenfeld [143]).

In the following pages edge detectors are classified according to the approach adopted. First the generic underlying idea is described. Next a more concise description of how this basic idea was implemented is presented.

2.9.2 Filtering

There are a number of methods that can be considered as distinct implementations of a filtering approach.

2.9.2.1 Derivative approaches

It is a direct consequence of the definition that edges correspond to areas with a high value of the first derivative. The most basic approach is to take a derivative of the picture, and consider as edges the points where the derivative value reaches a maximum or exceeds a given threshold. As derivatives enhance the edge these methods are sometimes referred to as enhancement / threshold techniques. This approach has various problems. Firstly as the derivative is a high pass filter it amplifies the noise in the image. Secondly it needs the use of a threshold which is not always clear. Thirdly the localisation effect, if a simple threshold is performed edges can be marked with a variable width with the consequent loss of precision. Fourth, due to the blurring edges are marked with variable widths.

The maximum derivative is obtained in the gradient direction, defined as:

$$\nabla \mathbf{F}(x,y) = \sqrt{\left(\frac{\partial \mathbf{F}}{\partial x}\right)^2 + \left(\frac{\partial \mathbf{F}}{\partial y}\right)^2} \qquad (1)$$

where $\nabla \mathbf{F}(x,y)$ is the gradient for the grey level function of the image F in the pixel (x,y) and $\frac{\partial \mathbf{F}}{\partial x}$ *and* $\frac{\partial \mathbf{F}}{\partial y}$ are the directional derivatives in the x and y directions respectively.

For a digital picture, first differences are used instead of the partial derivatives. The partial derivatives are then calculated by

$$\Delta_x(x,y) = f(x,y) - f(x-1,y) \qquad (2)$$
$$\Delta_y(x,y) = f(x,y) - f(x,y-1) \qquad (3)$$

To measure the central differences at the same point, the differences are measured over the 45° and 135° diagonal and are calculated by

$$(\Delta_+ f)(x,y) = f(x+1,y+1) - f(x,y) \tag{4}$$

$$(\Delta_- f)(x,y) = f(x+1,y+1) - f(x,y) \tag{5}$$

this is equivalent to the image being convoluted with the following masks:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \tag{6}$$

The result can be computed by the square root of the sum of the squares as in the gradient. This operator is known as the Roberts gradient operator[140].[1]

To avoid the computational complexity of the square root, it can be replaced by the sum of the absolute values or by the maximum of both values. Each of these forms present a slightly different behaviour. A comparison by Abdou and Pratt [2] will be referred to later.

This filter has a fast execution speed, due to the small amount of operations. It is one of the most precise. due to the small window used (2x2). Uncertainty of the position of the edge pixel exists due to the even size of the square used. It also presents a remarkable noise sensitivity.

The partial derivatives in the x and y directions could be estimated in a 3x3 window using differences centred on the point P(x,y). These formulae are known as the Sobel Operators and are probably the most widely referred to

[1] This corresponds to the technique described earlier. with shifts in the 135° and 225° directions.

edge enhancement scheme. They are defined by the x and y components of the derivative as

$$S_x = [f(x+1,y-1) + 2f(x+1,y) + f(x+1,y+1)] - [f(x-1,y-1) + 2f(x-1,y) + f(x-1,y+1)] \quad (7)$$

$$S_y = [f(x-1,y+1) + 2f(x,y+1) + f(x+1,y+1)] - [f(x-1,y-1) + 2f(x,y-1) + f(x+1,y-1)] \quad (8)$$

The result can be computed by the square root of the sum of the squares as in the Roberts operator.

These correspond to image convolution with the following masks

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (9)$$

The Sobel operator is slower due to the larger number of operations performed. It also less sensitive to noise as it can be considered as the composition of a smoothing and a highpass filter.

A different set of masks was proposed by Prewitt [138]:

$$\begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad (10)$$

2.9.2.2 General case

A large set of masks have been proposed [137] for edge detection. From the collection of convolution masks it can be assumed that any kind of vertical and

horizontal difference pattern or mask will give an edge enhancement operator. It is enough that the convolution originates a null (or small) value in the absence of a discontinuity, and otherwise a strong response, preferably proportional to the value of the discontinuity. This is generally true and these operators will present properties that differ between them.

One example of a non square mask is presented by Adhami [3]. He describes a cyclic ring of integers modulus 7 that have edge detection properties. This mask forms a 2x3 matrix.

The weighting coefficients and size of the mask will determine the frequency characteristics of the filter. In general it can be said that the larger the mask the more robust to noise it will be. It also requires more computations and will be less sensitive to fine details. This has lead to the suggestion of using composite approaches [128] [13] and Scale Space approaches [100][101].

The majority of masks presented so far have a square shape. Generally, the shape nearest to a square or circle will give better isotropy but less sensitivity to particular edge orientations. One way to improve the sensitivity to particular orientations or characteristics is to change to a rectangular, cross, or other shape. A long and narrow mask will present different noise sensitivity and detection characteristics in different directions. Examples of such masks are present in Hales operators (from [131]).

Davies [30] states that masks that have the same size but different coefficients correspond to different filters and thus will have different frequency responses. Thus masks are not interchangeable but, depending on the characteristics of

the signal and noise. application dependent. The choice of a particular mask should be related to a particular noise model [157].

2.9.2.3 Zero Crossings

One of the problems of the above approach is that a simple threshold will give thick edges and include more values than just the maximum. The criteria for the maximum could be incorporated in the edge detector. Since the maximum of the first derivative is a zero of the second derivatives, the second derivative can be used to detect the maximum. The edges are located now in the zero crossings of the second derivative. This could be done using the Laplacian operator, defined as:

$$Lap\, f(x,y) = \nabla^2 f(x,y) = \sqrt{\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}} \qquad (11)$$

Common masks employed to calculate the digital Laplacian are shown below

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \qquad (12)$$

$$\frac{1}{3}\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad (13)$$

When the image is noise free the zero crossings of the Laplacian of the image gives a very good edge map.

The main disadvantages of this type of filter are the bias that in some situations exists between the zero crossings and the position of the edges and. in the case of a noisy image, the creation of phantom edges. [116] [39].

A variation to this approach, suggested by Marr and Hildreth [108] uses the Laplacian of the Gaussian (LoG) as a way of reducing the noise sensitivity. They argue that this filter is the optimal trade-off between conflicting localisation requirements, corresponding to the spatial and frequency domains.

Marr and Hildreth first convolute the image with a Gaussian function. The Laplacian of this image is taken and edges marked at the resulting Zero Crossings. The commutative and associative property of the convolution permits the order of the operations to be changed and is the same as convoluting the image with the Laplacian of the Gaussian function. The resulting filter is defined by

$$\nabla^2 I(r) = -\frac{1}{\pi r^4}(1-\frac{r^2}{2\rho^2})e^{\frac{r^2}{2\rho^2}} \qquad (14)$$

where ρ is the standard deviation of the Gaussian used and r the distance to the centre of the Gaussian.

This filter is separable which reduces its computational complexity [174].

Haralick [58] [59] also detects the zero-crossings from the Second Directional derivatives obtained from a facet model. The facet model principle states that the image can be thought of as an underlying continuous or piecewise continuous grey level intensity surface. This can be understood as a smoothing step. The Roberts and Prewitt operators can be easily derived from a facet model fitted to the image.

Zhou, Venkateswar and Chellappa [190] represent the pixel in a local window by a 2D auto-regressive model whose parameters are recursively estimated.

Due to the modelling assumption the directional derivatives are a function of the parameters estimated. This method includes a global dependence in the sense that the estimated parameters include information prior to the position being processed in the raster scanning model. Also the derivatives are computed locally and thus the method can be termed as a local method.

2.9.2.4 Template Matching

A different approach, Template Matching, gives results which have similarities to the algorithms described so far. In the template matching approach the image is locally matched with a set of templates. The templates represent ideal edges with different orientations. The template producing the highest correlation determines the edge magnitude at that point and the edge orientation is assumed to be that of the corresponding template. This approach was due to Prewitt [138]. Her masks are obtained by rotating a kernel, originating a set of 8 templates corresponding to the 4 orthogonal directions and the 4 diagonal directions. Since the negation of a mask is also a mask, computation will only need to be done for half of the masks. The Kernel proposed is

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix} \tag{15}$$

Different sets of masks have also been proposed by Prewitt [138] and Kirsch [84].

$$\begin{bmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{bmatrix} \tag{16}$$

Robinson [141], Nevatia and Babu [122] (who use a set of 5x5 compass template masks) and Meer [112] .

Frei and Chen [42] proposed a different approach. They use nine mutually orthogonal masks. Since each mask is linearly independent from each of the others, they form a space within which we can define a projection criteria. The criteria of acceptability of an edge from the absolute value to the projection in the space is defined by the masks. They argue that this will allow more fine edges to be detected.

$$W_1=\begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \quad W_2=\begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_3=\begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \quad W_4=\begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

$$W_5=\begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \quad W_6=\begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix} \qquad (17)$$

$$W_7=\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \quad W_8=\begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

$$W_9=\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Some mask can be defined by continuous functions. An examples of such a mask was that proposed by Argyle[5]:

$$Arg(x) = \begin{cases} \frac{1}{\sqrt{2\pi}K} e^{-\frac{x^2}{2}k^2} & x > 0 \\ -\frac{1}{\sqrt{2\pi}K} e^{-\frac{x^2}{2}k^2} & x < 0 \end{cases} \quad (18)$$

Macleoad[104] presents a different function, consisting of the difference of two inverted exponential functions weighted by a third exponential. Abdou [1], uses a 10x10 mask to reduce computational complexity of the above formulae.

Kadar [79] uses Latin Squares for the mask definition. An ordinary Latin Square can be defined as a square array of (typically but not essentially) integers, such that each integer appears once only in each row and column.

Patton [128] proposes the Rule Based Composite Gradient Edge Operator (RBCGEO) which interprets and integrates the responses of three operators each designed to extract edge information at a specific spatial resolution. Three weighted templates (3x3, 5x5, and 7x7) are used. A rule based system is used to derive the magnitude. direction and confidence of the edge from the responses of the templates. This method was latter developed by Brazakovic. [13]

Weschsler and Kidode[181] define a different edge operator that can gather its data from windows of different size and shape. The edge enhancement step is

also a difference scheme followed by thresholding, thinning and noise removal.

2.9.2.5 Optimal filter approaches

Another approach defines the optimal shape of the filter that maximises a given criteria. The first operator following this approach was developed by Canny,[18] [17] in 1983.

Canny's maximises the following criteria: good detection, good localisation and only one response to a single edge. The filters obtained are defined by a set of six parameters, and are of the general form

$$C(x) = e^{\alpha x}(a_1 \sin \omega x + a_2 \cos \omega x) + e^{-\alpha x}(a_3 \sin \omega x + a_4 \cos \omega x) \quad (19)$$

where a1 to a4, α and ω are constants obtained by an optimisation process relative to the optimal criteria defined. For step edges the filter obtained was very similar to the first derivative of the Gaussian. Canny to avoid the computational complexity of his solution. uses the first derivative of the gaussian which scores 80% of the performance of the optimal filter. This more effective filter is usually referred to as the "Canny operator". A slightly refined version. using a more complex way to compute the derivative, of this operator due to Fleck [161] is described in Appendix A as it was implemented as described in chapter five.

Spacek [158] formed a performance measure combining all three of Canny's quantitative measures in order to simplify Canny's equations.

A different optimal filter was obtained by Deriche [34] [33] . Using Canny's criteria, he allows the filter to be of infinite size and obtained a function of the form

$$D(x) = -e^{-\alpha|x|}\sin \omega x \tag{20}$$

and derived the solution as an Infinite Impulse Response filter which allows a recursive implementation. This filter is quicker then the ones defined by Canny, and. as for Fleck's filter, a detailed description is in Appendix A.

Modestino and Fries [116], Sarkar [147] and Manickam [105] also proposed recursive filters.

Castan et al [19] shows that the optimal edge detection filter for high SNR, good localisation, precision and uniqueness of the maximum is a symmetric exponential filter. The filter obtained is

$$f(x) = a\ b^{|x|} \text{ with } \quad a = -\frac{\ln b}{2} \quad \wedge \quad 0 < b < 1 \tag{21}$$

where a and b are constants.

Barlaud et al [7] uses a Recursive biOrthogonal Wavelet Transform to compute optimal edge detection filters which yields multiresolution analysis.

Petrou and Kittler [134] extended Spacek's work and derived optimal filters for ramp edges of various slopes.

2.9.2.6 Mathematical Morphology

Mathematical morphology is a mathematical field which was formalised in the eighties by Serra [149]. Morphological edge detection was introduced by Lee et al [97] . A generalisation of this was proposed by Yoo [185]. It has been used for ridge detection by Hertz [65] . An application for medical imaging is described by Peng et al [132] .

2.9.2.7 Median filtering

Median filters were proposed for edge detection by Pitas [136] and Bolon [11]. A combination of a median filter and a linear filter is described by Neuvo [120] .

2.9.2.8 Other filtering approaches

Amini [4] models the picture as the output of a two dimensional all pole causal sequence with a Quarter Plane and Non-Symmetrical Half Plane support regions. The estimated parameters are used to approximate the 2nd directional derivative. Auto regressive models are also used by Zhou et al [190].

Other filtering techniques have been proposed as Gabor wavelets odd functions [113] , the E filter [146] . Linear Model [81] , Stack filters [185] or adaptive FIR filters [57].

2.9.2.9 Decision criteria

Most of the above methods give as the result an edge enhanced map. which has assigned to each point in the image a value characterising its "edginess".

A criteria for the acceptance of the point as an edge must be defined. The quality of the resulting edge map is dependent of this decision criteria.

The most widely mentioned and simple decision method is to establish a single threshold. All points with a higher value than the threshold are accepted as edges, otherwise they are marked as non-edges. The main role performed by the threshold is the selection of the most 'abrupt' discontinuities. This selection allows the points that result from small local variations, namely due to noise, to be discarded. This threshold is easy to establish if the amplitude of the edges are quite different from the average values that result from the processing of the noise. In this case, the histogram of the edge enhanced image will present a clear bi-modal characteristic and the selection is clearly made by the minimum value between the two maxima. This criteria gives thick edges. A simple threshold will always provoke a conflict between the number of assigned edges and the thickness of the edges. A simple criteria will also limit the sensitivity of the operator, as soft edges will be rejected along with the noise induced points[96].

Some of these problems can be solved by the use of the maximum within the lines marked in the edge enhanced map. This criteria is known as non-maxima supression [142]. Also a double threshold can be used. In this case two thresholds values are defined, say, t_{low} and t_{upp}, with, $t_{low}<t_{upp}$. All the points greater than t_{low} which belong to a curve that contains at least one point greater than t_{upp}, are marked as edge points. This criteria is used, for instance, to select the strongest zero crossings by Canny [17] and Fleck, to select the most significant edges avoiding the spurious lines resulting from the noise[38].

edge as being a true edge, and P(TE|AE), giving a true edge of being an assigned edge. The best threshold is the one that minimises the difference between these two values.

This criterion however presents some disadvantages considering that we must (i) known "a priori" the position and localisation of the edges, and, (ii) we must compute that criteria for all the edges, which is time consuming. This method, although proposed as a threshold criteria has been used, as a comparison criteria by some authors [157]. Another method is the one used by Fleck, where the thresholds are fixed as multiples of the standard deviation of the processed image in edge free areas. Although requiring the definition of an edge free area, this criteria is more easily implemented than the previous one.

2.9.3 Model fitting

The most representative edge profile is the step form. an ideal edge can be assumed as a step discontinuity in a particular region of the image. Another approach is based on the idea of how closely this model fits a given image neighbourhood, decisions then being made according to the quality of the fit. This approach was proposed by Hueckel [72]. Hueckel used a step edge defined in a co-ordinate system that lies in the centre of a window. He used a window with a circular shape. and typically consisting of 72 points (to minimise the errors due to the radius quantification). The method fits a step function to a window and according to the fitting result decides if it is acceptable or not. This implementation of the approach is computationally expensive due to the need to calculate Fourier series of some of the functions.

A computationally lighter approach was defined by O'Gorman [124]. He used Walsh functions instead of the Fourier series for the expansion of some intermediate functions. This implementation is also described in detail in Appendix A. Merõ and Vassý [114] looked to Hueckel's operator as a template matching method and reduced it to 2 matching patterns. Nevatia [121] used a subset of the Hueckel basis. Hummel [73] generalised Hueckel's results using a different expansion.

A different edge model. in which a monotone varying edge can be defined, was used by Shaw [152]. This can be optimally matched for an image signal within an operator window of dimensions NxN. Wahl went on to derive the general case in 1976 [177].

2.9.4 Stochastic and statistical related methods

To decide where an edge of a given orientation exists in a window, one can bisect the window at that orientation thus creating two sub-regions, calculate the appropriate features and decide as to the presence of an edge based on some selected threshold. Yakimovsky used this approach [183] . The decision concerning the exact location of the edge was left to a region grower. This is one of the nearest approaches to some of the definitions presented at the beginning of the chapter. However, examples by Vernon[174] show a relatively poor performance.

Amlan Kundu [93] described a method where step and linear edges are detected as a statistical classification problem. based on two characteristics: Near and around the step and linear edges. the pixel. when classified into two

nearly equal groups, displays markedly different average intensity values and the member of each group show strong spatial correlation.

Bovik and Amunson[12] use median comparisons instead of average values.

2.9.5 Residual analysis

After the smoothing operation, the difference between the result and the original image has the characteristics of noise in areas away from features. This can be used for feature detection and has been used by Chen, Lee and Pavlidis [23] . A cost minimisation approach is used by Tan [166] .

2.9.6 Neural networks

Neural networks can be used for edge detection or as edge improvement systems. Sinchak [155] proposes the use of an Artificial Neural Network to improve edge measurement in noisy images. It is used to adjust the edge measurement based on the neighbouring edges.

Neural networks are used by Spreeuwers[159]. A 49-8-1 two layer feed forward neural network is trained with examples of edge and non-edge patterns using the back propagation learning rule. Kerr[83] proposed a method for training standard feed forward back propagation networks using fuzzy label vectors. The neural network is used as an edge enhancement operator which is thresholded when defuzzifying occurs. Results presented show that the neural network can be successfully trained and subsequently operated as an

edge detector that is as least as good as the Sobel and Canny operators. The training phase is completely independent of any real image and is done with a basis set of 256 binary windows as the input.

Terry et al [167] gives a simple method for the edge detection training problem. Not only can Neural Networks be trained to detect edges, they can also be designed from scratch without the prior necessity of training. Comparisons between both cases are presented. A neural network using multi-state ADALINES[2] is presented by Paik [126].

2.9.7 Other approaches

Several different approaches are possible using distinct smoothing techniques. such as anisotropic diffusion[123]. Anisotropic Diffusion operates by repeatedly filtering the image function with a smoothing kernel of small support. thereby producing a sequence of diffused image functions of successively lower resolution. In order to retain the strength and correct location of edges, the "smoothing power" of the filter kernel is made to depend (inversely) of the magnitude of the image function gradient in a heuristic fashion. At some stage in the iteration filtering process remarkably impressive edges can be extracted by post-processing the diffused image function with a rudimentary local edge detector.

: Adaptive linear neurons

2.9.8 Global methods

The main criticism made about the above techniques is that the context of the edge or the scene information is not used. Although some techniques related the points in a neighbourhood with the other points in the image, global information is not present in any of them. Several methods have been proposed that include the whole image, instead of focusing the attention in small areas or windows.

Kelly [82] used heuristic planning for finding the edges in a picture, based on the edges of a smaller and less detailed version of the original.

Montanari [117] proposes a method which embeds the properties of a curve in a figure of merit. A dynamic programming technique is then used to determine the optimal curve in the picture with respect to the given figure of merit.

Martelli [110] reduces the problem to the search of an optimal path in a graph. using the A* algorithm. The structure of the graph is determined by the properties which the edge must have. In 1976 [109] he presented different work where the properties of an edge are contained in the figure of merit and the problem becomes the problem of minimising the given figure of merit.

Griffith [51] describes a system of programs for the detection of straight line edges in simple scenes. The program goal is to locate all the edges in scenes consisting of prismatic solids and to make fundamental use of problem

reduction techniques and a priori information derived from an explicit model of the situation or problem under consideration.

Eberlein [37] tried iterative methods for the edge detection. An iterative method to detect edges in a one dimensional signal is presented by Chen and Medioni [22]. The results can be combined in two orthogonal directions to give a two-dimensional edge map. A similar approach was done by Tewfik [168] .

Zhou [189] presents an AI structural approach based on the intuition as to the characteristics of an edge. Where an edge should be at the place where the grey level of the picture has great change and which has a long but narrow shape, i.e. it uses both grey level and structural information from the picture. The approach consists of several steps. First, the input image P(i,j) is transformed into a so-called edge image D(i,j). This can be carried out using a number of operators such as the Robert's operator. The second step transforms this image into the form of a labelled tree. Edges are then extracted reducing to the main branches.

2.9.9 Assessment of edge detection techniques

According to Vernon.

> "One point where there is a definite consensus in the computer vision community is that it is difficult quantitatively to evaluate and compare edge detection performance"[3].

[3] Automatic Visual Inspection and Robot Vision, page 106, Prentice Hall 1991

An empirical assessment is sufficient for the majority of applications. It is, however, desirable to have, "even if only to provide a rough guide to the relative performance of the detectors in particular circumstances" [174] a quantitative evaluation. However, as Pratt states

> "The only performance measure of ultimate importance is how well edge detector markings match with the visual perception of object boundaries" (W.K. Pratt, op. cit. page 543).

Effectively a human observer is able to discern and evaluate the quality of the response of the edge detectors by observation as a human observer is able to recognise objects and trace their boundaries. However similar situations are not easily discernible by a human observer. Effectively a human observer will be able to evaluate the shape and definition of the edges produced. but will have difficulty in grading similar situations. The problem is not only a direct consequence of the multiplicity and fuzziness of edge definitions, presented in the beginning of the chapter, but a consequence of the gap between the definition and objectives.

However, single properties can be measured allowing for a more reliable and accurate assessment. The following section will revise several criteria proposed and comparison performed.

2.9.10 Comparisons between edge detection algorithms

Several studies have been carried out into the comparison and assessment of edge detectors. Unfortunately, in the majority of cases, results from a particular figure of merit are only known for the methods that were presented by the author who proposed it.

Some edge detection 'figures of merit' have been proposed. However, they do not fully test all the aspects of the edge detection techniques. The figures of merit proposed are limited and can not be recognised as a method for the grading of edge detection algorithms. None of the proposed measures are able to characterise the edge shape.

Fram and Deutsch [41] have studied the effects of noise on various edge detector schemes. They used 36x36 test images divided into 3 vertical zones. in which they assign random selected grey levels to the exterior ones.

Pratt [137] uses a square array of pixels with a vertically oriented ramp edge at its centre. The edge parameters and noise level can be varied to generate test edges that are then processed by an edge detector to produce binary edge maps. The figure of merit is:

$$F = \frac{1}{\max(I_I, I_A)} \sum \frac{1}{1+\alpha d^2(i)} \qquad (22)$$

where I_I and I_A are the number of ideal and actual edge points. d(i) is the pixel distance of the ith detected edge and $\alpha=1/9$ a scaling constant to provide a relative penalty between smeared edges and dislocated edges. This figure of merit, probably one of the most common, has been criticised as it does not

reflect directly the characteristics of real edges. It also requires, in order to be representative of the behaviour of the edge, the detector to be used over several images with different contrasts and orientations.

Abdou and Pratt [2] [137] compare edge enhancement / thresholding techniques (Differential e template matching, and threshold decision), with regard to several characteristics (sensitivity, amplitude response, etc.). They define an idealised luminance edge as a plane ramp discontinuity. The ideal edge is described by its cartesian pixel co-ordinate, orientation angle, base amplitude, contrast, and slope.

Bryant and Bouldin [15] proposed two means for numerically evaluating edge detectors, designated as relative and absolute grading. Relative grading involves comparing an edge operator output to the consensus decision of the other operators. Absolute grading compares any operator to a manually constructed key for a target scene. It can be thought of as a measure of a real operator and an ideal operator. A common method of computing this similarity is to determine a correlation measure.

Kitchen and Rosenfeld [86] used local edge coherence. The measure is based on the continuation and thinness of the edges. The measure is defined as

$$E = \gamma C + (1 - \gamma)T \tag{23}$$

where c and T are respectively the continuation and thinness measures. and γ is a scaling constant. This criteria does not need a reference picture. as most of the figures of merit proposed by others authors.

Peli and Malah[131] show several performance measures, quantitative and qualitative (Type of contour, single or double edge, distortion). The algorithms considered are the Roberts algorithm, Hale's operators and the Rosenfeld Algorithm. One conclusion is that sometimes the Pratt figure of merit rating does not provide sufficient information about the performance of the edge detector tested.

Haralick [61] compares several edge detectors, namely Marr, Prewitt and the Directional derivative based on the facet model. Comparisons are carried out using the average and the probabilities P(AE|TE) and P(TE|AE).

Delp and Chu [32] present results with Pratt's and Kitchen's figure of merit. The spatial operators are used to generate an edge strength and direction maps. To these maps are applied a contour tracing algorithm. The most relevant aspect is the SNR insensibility of the Haralick second directional derivative operator (which uses 11x11 masks) to Pratt's figure of merit. Also the poor score obtained by Marr's zero crossing operator is quite remarkable.

Sousa [157] uses a generated checkerboard image with several levels of added white noise. The figure of merit is the same as that presented by Haralick.

Van der Heyden [66] proposes the use of grey tone test images, accompanied by binary reference images and a new figure of merit. Unfortunately he does not present extensive application results.

De Micheli [115] analyses two aspects of edge detection: accuracy of localisation and sensitivity to noise. The aim is to establish the best scheme for edge detection of usual indoor scenes. Two basic conclusions are presented: They think that a gradient scheme is superior to a zero-crossing scheme and given the amount of noise present in typical indoor images the exact shape of the filter used to regularise differentiation is not critical.

Kitchen and Malin [85] calculate, for a unit step edge, the gradient magnitude and direction reported by various simple differential edge operators as a function of the edge's actual orientation and offset with respect to the pixel grid.

Amini [4] presents a comparison between parametric and non-parametric edge detectors using step edges and an image of squares with gaussian noise.

Bernsen [8] considers an edge detector as the sum of the component edge-strength computations, edge localisation and derivative computation. He uses a modified version of the Haralick test. For test images a checkerboard image is used for objective comparisons and a surface mounted device is used for subjective comparison.

Fleck [39] presents a study on artificial features typically produced by Canny's, Marr's and Boie-Cox's Algorithms.

Venkatesh and Kitchen [173] classify different kinds of error that can occur in edge detection and then develop measures for quantifying these errors. The

four error types defined are: False negatives, false positive, Multiple detection and Localisation error.

However, as it was stated by Pratt, the 'performance measure of ultimate importance' is the visual analysis and such comparisons are presented by Bernsen [8] using a surface mounted device image, Pratt[137] using an image of several peppers, and Vernon [174] using an image of a set of wires.

2.9.11 Discussion

One question that always arises is "Is there a best edge detection algorithm?". For the majority of images, that present very low levels of noise and blurring, there are no sparkling differences in the edges detected by the different algorithms. They simply mark different edges.

A large number of edge detection algorithms have been proposed and every year new algorithms are being proposed. Some of the questions that arise from the definition of an edge have been presented. They were followed by a review of proposed edge detection algorithms and comparison that have been described in the literature. The remainder of this chapter is concerned with the topic that forms the basis of the proposed approach to solve some of the problems associated with edge detection, namely the use of artificial neural networks.

2.10 Artificial Neural Networks

2.10.1 Introduction

Neural networks have been a research topic for nearly half a century. They were first developed as a model of the human brain, with the purpose of modelling and understanding the principles on which the human brain works. Another reason was the wish to build machines that are capable of performing complex tasks. Neural networks are models for cognitive tasks. One of the first neural networks, known as the *'perceptron'*, was developed as a simplified model of the biological models of processing sensory information, or perception systems [119]. Although some limitations of the 'perceptron' slowed the development of neural networks for several years the definition of an effective learning rule for complex perceptrons expanded their application areas.

Neural networks terminology is based on the nervous system. Basically the nervous system is composed of neurons, which are the main cells from which nerves and the brain are made. These cells develop extensions. called dendrites and axons. which allow them to send and receive information from and to other neurons or muscles. by the joints, called synapses. Artificial neural networks, consist of independent 'cells', called neurons or nodes. which exchange information from and to other neurons by links. the synapses. of variable strength. or synaptic strength.

Using a less biological approach. artificial neural networks are. as defined by Müller.[119]

'directed graph with the following properties:

1- a state variable n_i is associated with each node i

2- a real value weight w_{ik} associated with each link (ik) between two nodes i and k

3- A real valued bias v, associated with each node i

4- A transfer function $f_i(n_k, w_{ik,}, (k \neq i))$, is defined for each node i, which determines the state of the node as a function of its bias, of the weights of the incoming links, and of the states of the nodes connected to it by these links.'

(op.cit. , page 12)

The most common transfer function is a sigmoidal function, which usually takes the following form

$$S_\gamma(x) = \frac{1}{1+e^{-\gamma x}} \tag{24}$$

This function is plotted in figure 6 It is plotted as Sig(x,y) to allow a direct comparison with figure 13. A linear approximation,

$$L(x) = \begin{cases} -1 & x \leq -1 \\ x & -1 < x < 1 \\ +1 & x \geq 1 \end{cases} \tag{25}$$

or a hard threshold

$$thr(x) = \begin{cases} -1 & x \leq -1 \\ 0.5 & -1 < x < 1 \\ 1 & x \geq 1 \end{cases} \tag{26}$$

can also be used. The later is usually associated to pattern recognition applications, where a binary response is required. The linear approximation is used either with binary or continuous data and its principal advantage arises

from its computational simplicity. This function can be used in different stages of the development of a neural network application.

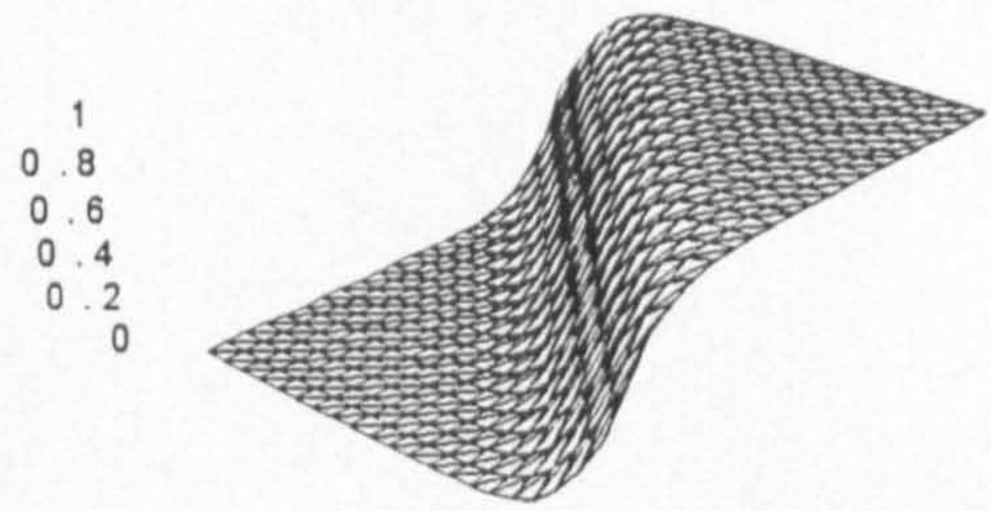

Figure 6: Sigmoidal function Sig(x,y)

The main difference between artificial neural networks and traditional methods. resides in their development. Artificial neural networks basically reproduce functions on the basis of their measured behaviour. Their main advantage and uniqueness arises from the way in which that relation is produced. Artificial neural networks are not programmed but taught. They learn from 'experience' and are able to adapt themselves in order to reproduce the relation between the input and output sets of data.

Learning can be performed on an unsupervised or supervised basis. An unsupervised learning process requires only input vectors to train the network. During the training the network weights are adjusted in an ordered way according to some defined figure of merit. In a supervised learning process. input/output pairs are presented to the network. and. according to a defined law. they will change their own weights in order to be able to reproduce the correct output vector when an input vector is presented.

Although still having some relatively undefined aspects of development. neural networks have been applied successfully to a wide range of problems. such as pattern recognition [175] [28] [176] and automatic control [179]. Several network structures, none perfect. have been proposed by different authors. These are reviewed in the next sections.

2.10.2 Unsupervised Neural Networks

The most described unsupervised artificial neural network is the Kohonen [89] Self Organising Map (SOM). It mimics some of the organising properties of the brain neurons. It is particularly useful for real world applications, where no 'a priori' knowledge is known about the organisation of the input data. It is trained through sequential presentation of continous-valued input vectors. without any specification regarding an output. It consists of a set of input nodes. connected to all output layer nodes. which , in turn, are locally interconnected. The weights of the links between input and output nodes are initially set to small random values. It also requires the definition of the neighbourhood of each node. which slowly decreases during the learning process. Once a suitable number of input vectors has been presented. weights will specify cluster centres that sample the input space. As a consequence. the weights will be organised in such a way that close nodes will be sensitive to similar inputs.

2.10.3 Supervised networks

2.10.3.1 Hopfield Net

The first supervised network to be described is the binary Hopfield Net[71] [31]. Its most common application has been within the context of associative or content addressable memories. An information storage device is called an associative memory if it allows for recalling of information through a partial knowledge of its contents, but without knowing its storage location. Another application of the Hopfield network consists in its use as an auto-associative memory. When a set of patterns has been learnt, if a noisy or incompletely defined example its presented, it can recover the initial pattern.

The basic structure of a Hopfield network is presented in figure 7. It consists of a single layer of processing units. Each unit has a binary activity value or state. All processing units are fully interconnected by directional links. In the updating procedure each neuron is set to the signal of the weighted sum of its inputs. The weights are previously calculated, using a given set of equations. It allows the addition or subtraction of patterns from the defined set, after the definition of the network without recalculating the full set of weights to be remembered The updating process could be performed in sequence, and repeated until a stable state is attained. Although not being a stability necessary condition, mathematical analysis shows that the network is able to attain a stable state once the weights of coupled links are equal.

Hopfield networks have significant limitations since at times they evoke spurious states that were not included in the original learning set. The number of spurious states could be reduced by an adequate unlearning process.[69].Another of the limitations arises from the small number of

patterns stored, typically in the order of 10% of the number of neurons [69] [179].

Hopfield networks can be generalised if the deterministic evolution law is replaced by a stochastic law. In this case, instead of binary values, the network will give the probability associated with the response taking one of the values ±1. This type of network is commonly designated as a 'stochastic neural network [119]. A further generalisation are Boltzmann machines which also have hidden layers of neurons and form a general computing machine. Due to the stochastic behaviour of these networks, convergence to the same solution could not be achieved. particularly if there is more than one valid solution to the problem.

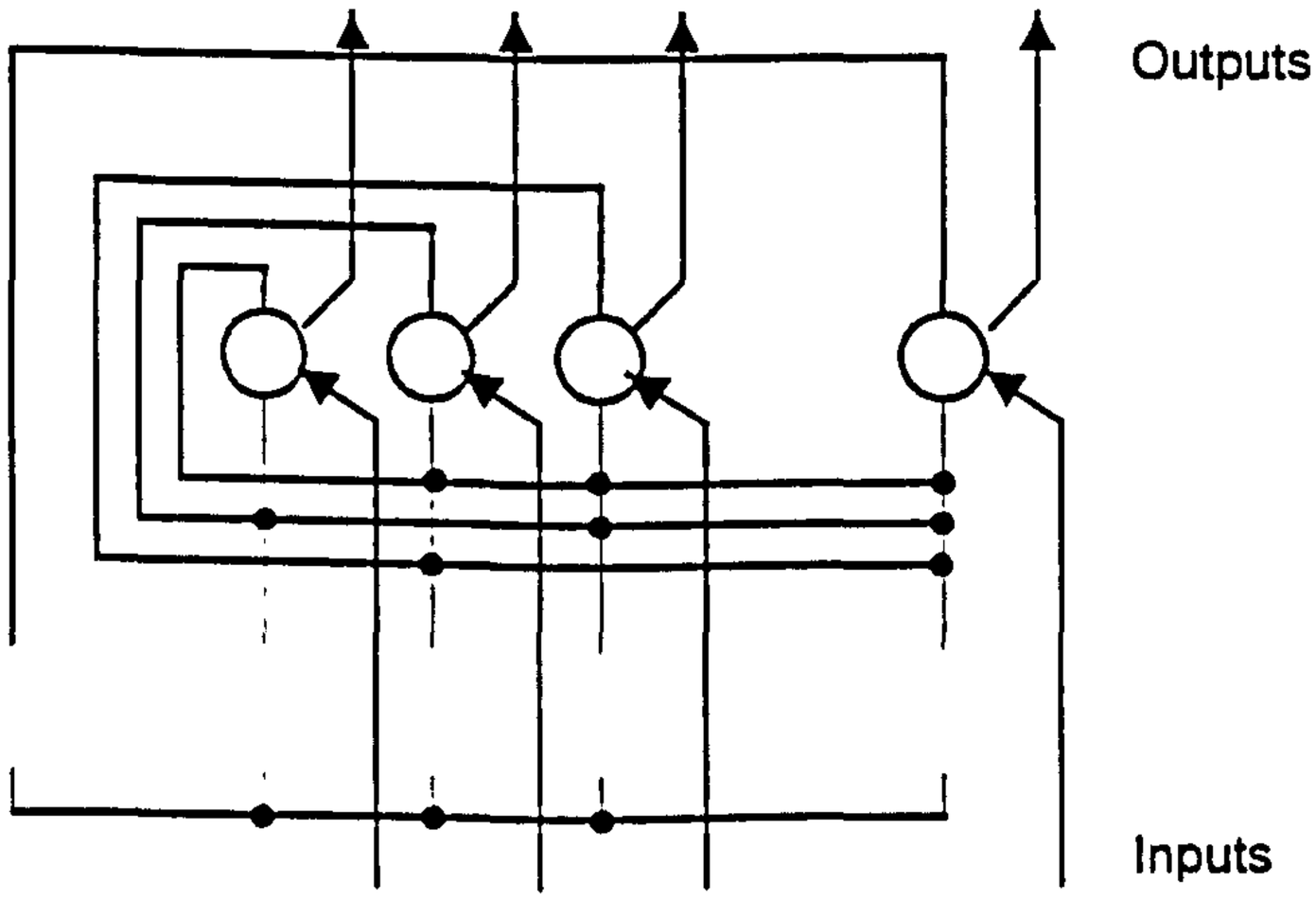

Figure 7:Hopfield Network

2.10.3.2 Hamming Nets

Another type of network is the Hamming net [99] .They implement an optimum classifier. which is the Hamming distance. The Hamming distance is

the number of bits in the input that do not match the corresponding exemplar bits when the signal is sent through a binary symmetric channel.

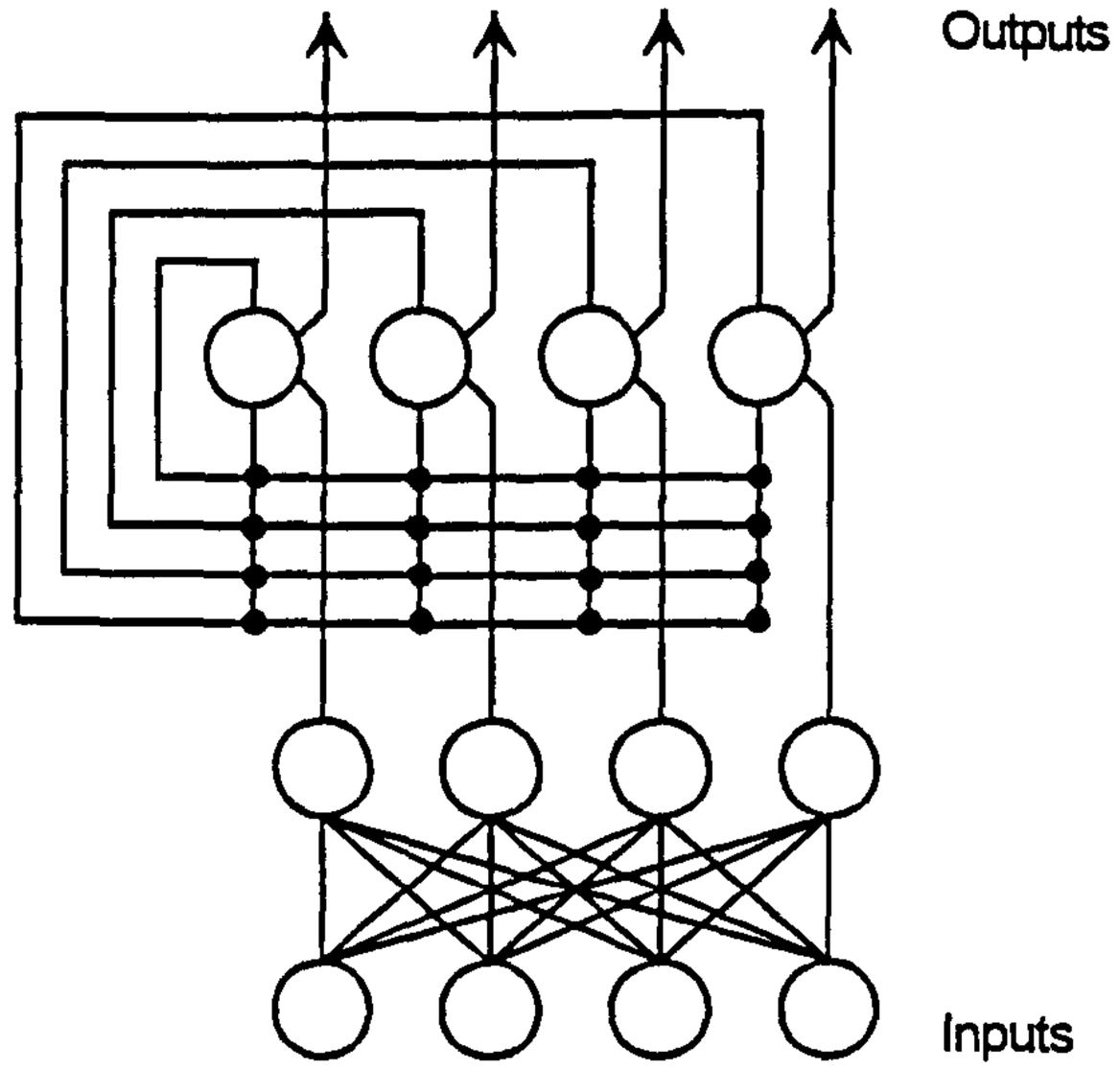

Figure 8:Hamming Network

Figure 8 represents schematically a Hamming Network. This type of network is a mixed network with two subnets, the first of which has a perceptron like structure (see section 2.10.3.3) with two fully interconnected layers. In this net the weights and thresholds are set in such a way that the outputs in the middle layers are equal to N minus the hamming distance to the N patterns used. The second subnet has the function of determining the maximum of these scores.

The Hamming net has some advantages over the Hopfield net. It implements the optimum error classifier when bit errors are random and independent. Hence allowing for equivalent or better performance than that achieved using the Hopfield net [99]. It also requires fewer connections than the Hopfield net. Another advantage of the Hamming Net is that it does not suffer from spurious output patterns, unlike the Hopfield networks.

2.10.3.3 'Perceptron'

Perceptrons are an example of feed forward layered neural networks, where information flows in one direction between several distinct layers of neurons. The output of each neuron, represented in figure 9, is a function of the multiple inputs that it can accept. The simplest case, usually called 'single layer perceptron', consists of two neuron layers fully interconnected, designated by input and output layers, and from which the first only accepts inputs without processing them. The perceptron described is adequate for simple classification problems.

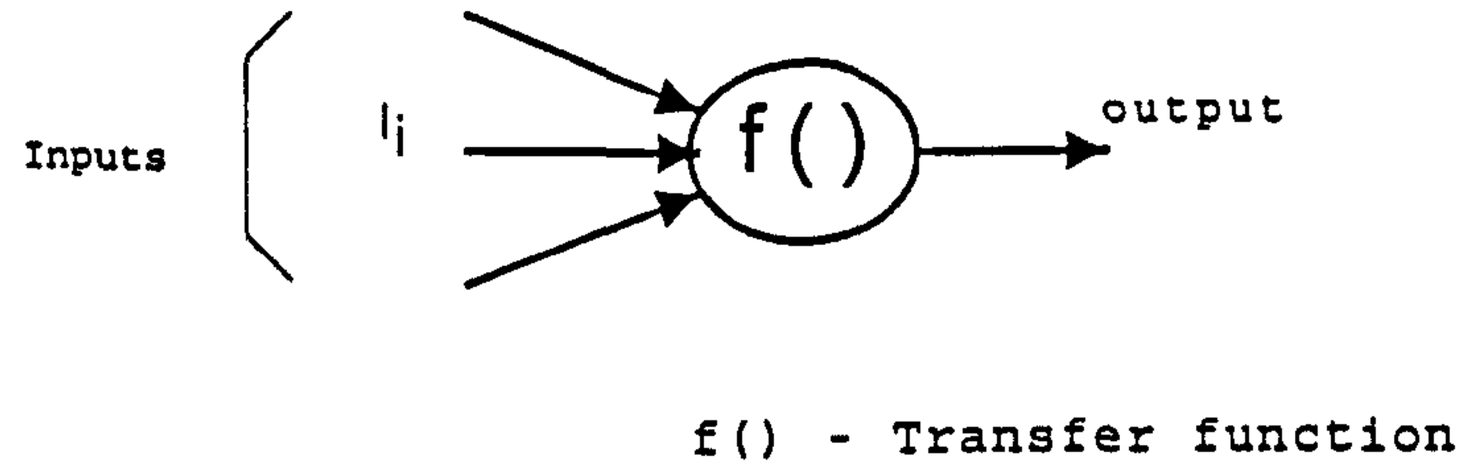

Figure 9:Basic perceptron unit

This limitation has been largely overcome with the introduction of an effective learning rule for a multi-layer case. This includes one or more hidden layers between the input and output layers as represented in figure 10 .

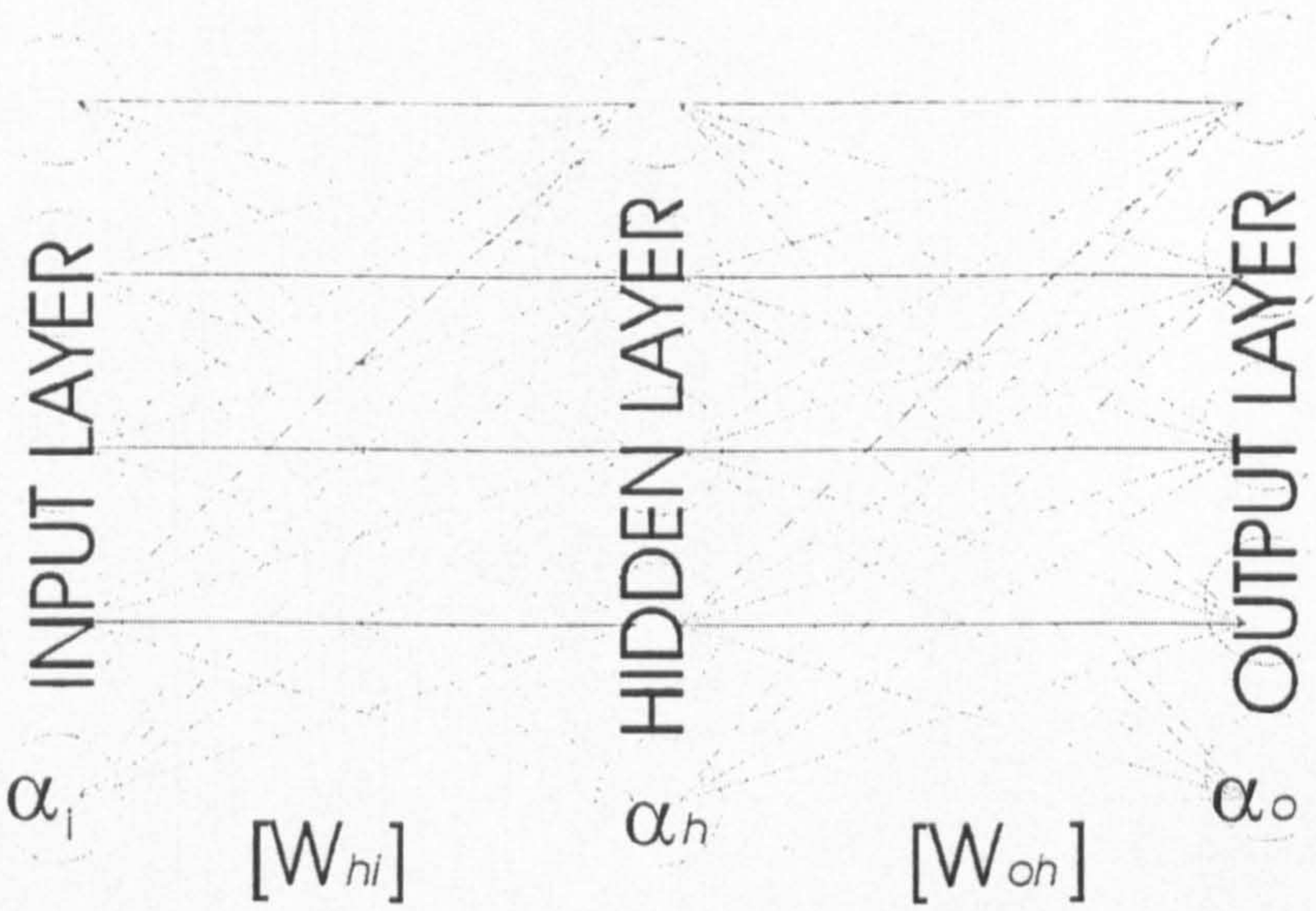

Figure 10:Multi-layer Perceptron

This was a major step in the history of artificial neural networks. Today, this learning rule, known as backpropagation (or one of its variations), is the most common neural network training method. In fact, according to Wasserman [179], it covers more than 85% of published applications. Indeed back propagation is suitable for almost all applications as long as input/output pairs can be defined.

Backpropagation is a mixed blessing. There are at least three arbitrary coefficients for which no rule, even approximate, is known and given an unfortunate choice could cause the convergence to slow or even stop. This phenomena is usually referred to as 'paralysis'. It also presents long training times. While in theory training need only be done once, system development inevitably requires a certain amount of iterative optimisation, particularly when selecting the appropriate features from the data set.

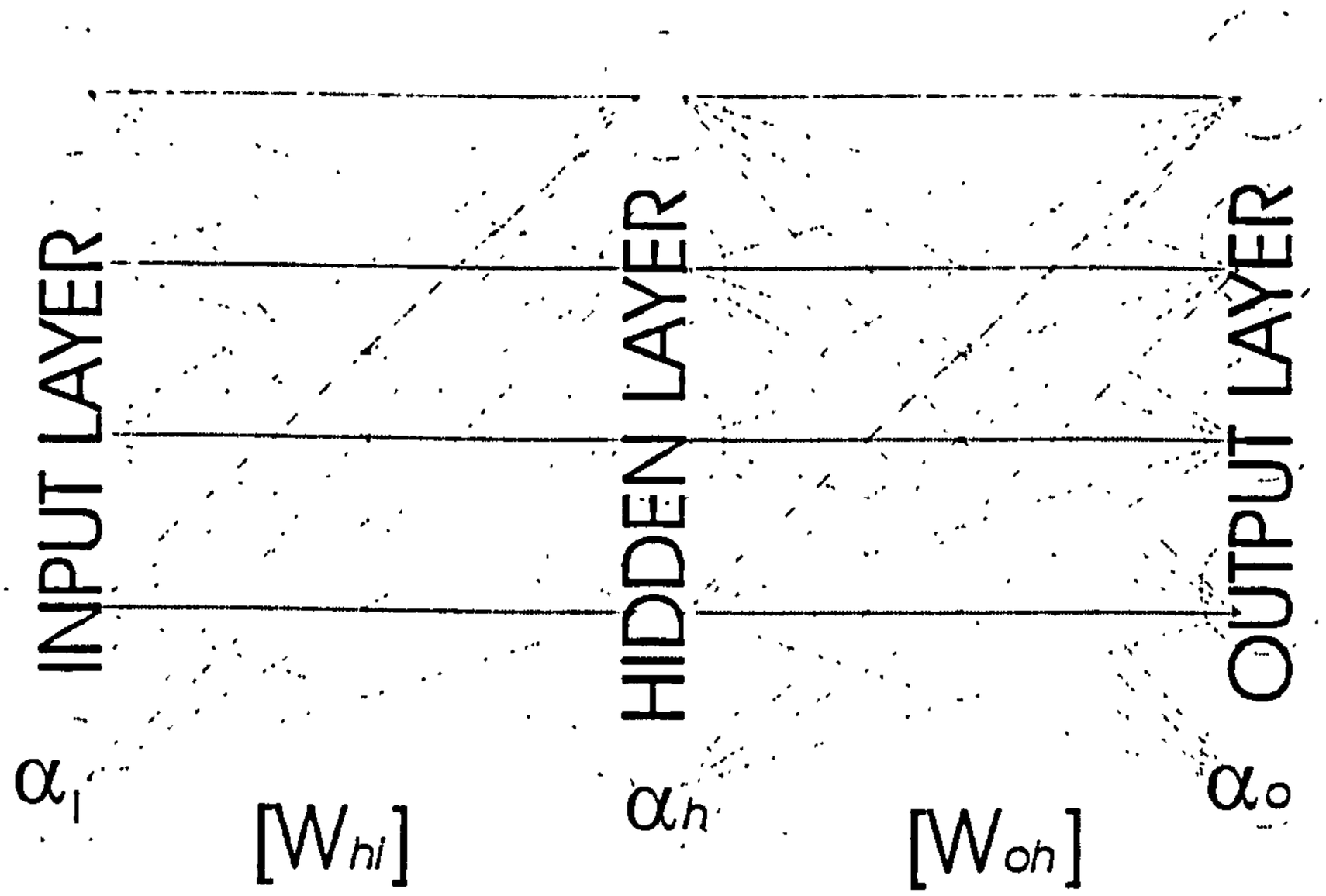

Figure 10:Multi-layer Perceptron

This was a major step in the history of artificial neural networks. Today, this learning rule, known as backpropagation (or one of its variations), is the most common neural network training method. In fact, according to Wasserman [179], it covers more than 85% of published applications. Indeed back propagation is suitable for almost all applications as long as input/output pairs can be defined.

Backpropagation is a mixed blessing. There are at least three arbitrary coefficients for which no rule, even approximate, is known and given an unfortunate choice could cause the convergence to slow or even stop. This phenomena is usually referred to as 'paralysis'. It also presents long training times. While in theory training need only be done once, system development inevitably requires a certain amount of iterative optimisation, particularly when selecting the appropriate features from the data set.

Back propagation, despite all its problems, remains a highly effective paradigm. In difficult applications, where the input/output relationships are non-linear and/or involve high order correlation among the data, it will produce accurate results. The main disadvantage is its slow training , although in comparison to RBF networks presented next, it could be partially redeemed by its more efficient computation at application time.

2.10.3.4 RBF networks

In very recent years a wide range of similar paradigms have been developed. These can be grouped under the generic name of basis functions techniques. Among them the radial basis function paradigm is the most used [179].

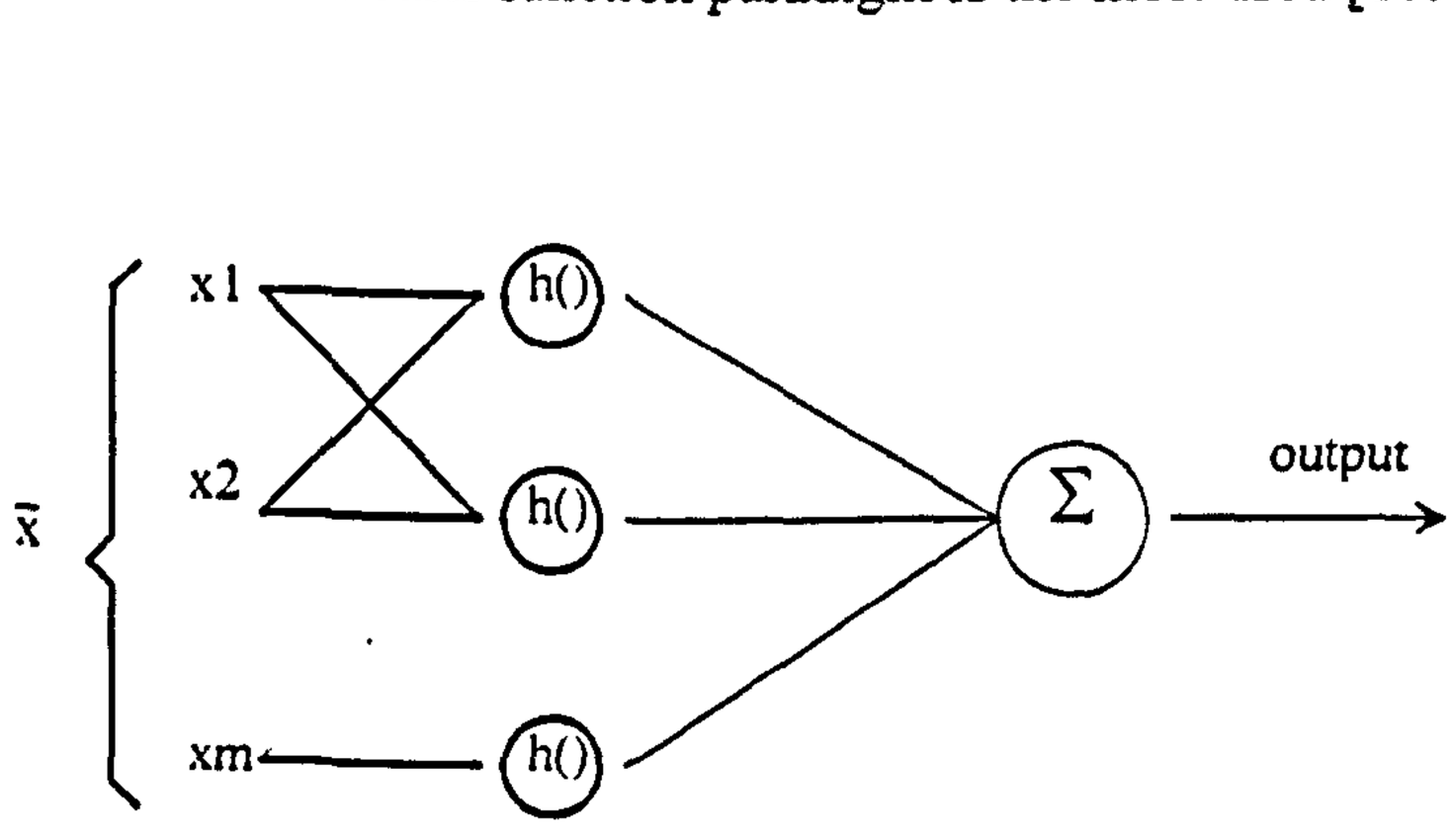

Figure 11:RBF Network

Figure 11 presents such a network. The basic hidden neuron shape is shown in figure 12.

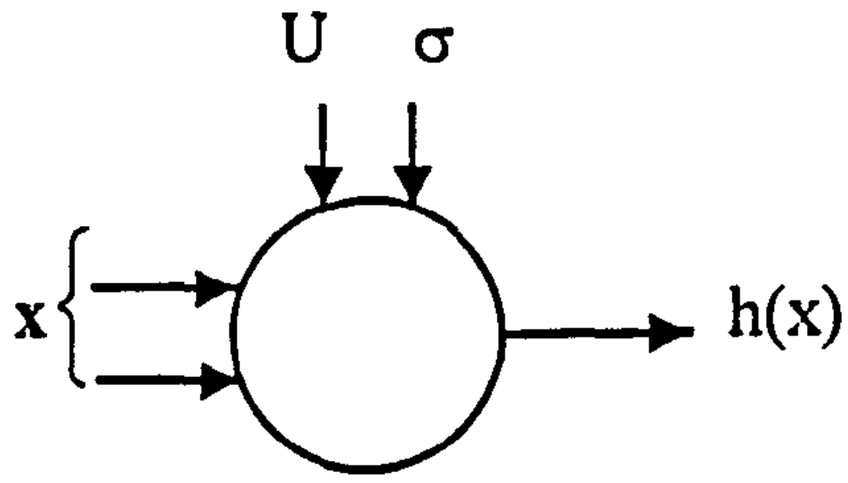

Figure 12: RBF basic neuron

The inputs are applied to all neurons in the hidden layers. Each hidden layer neuron computes the following exponential function:

$$h_i = e^{\left[\frac{D_i^2}{(2\sigma)^2}\right]} \tag{27}$$

where

$$D_i^2 = (\bar{x} - \bar{u}_i)^T(\bar{x} - \bar{u}_i) \tag{28}$$

and u_i is the weight vector of hidden layer neuron i. A two dimensional plot of this function is shown in figure 13.

This network requires a hidden layer neuron for each training vector. The output neuron produces the linear weighted summation of these. The output will only give a relevant response to an input over a range of values called the receptive field of the neuron, the size of which is determined by the value of σ

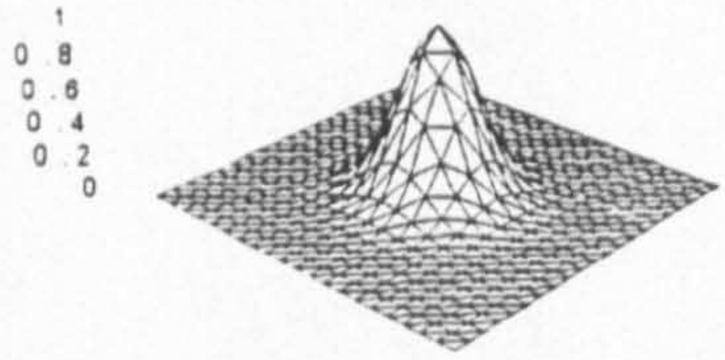

Figure 13:RBF transfer function

The location of the centres of the receptive fields is a critical issue and there are many alternatives for their determination. A centre and radius corresponding to each hidden layer of neurons could be located at each input vector in the training set. Because training vectors tend to occur in clusters this method will, in general, result in more hidden layers neurons than necessary. The result would be long training times and slow operation during reference, due to the large amount of computation required. Once the centres and sigmas have been chosen, the output layer weight can be optimised by supervised training.

Basis function networks train rapidly, since training is only used as a fine tuning, and part of the information is presented in a partially ordered form. Also they do not exhibit the training pathologies present in the backpropagation algorithm. They have one major disadvantage. After training they are generally slower to use, requiring more computation to perform a classification or function approximation. They also require a substantial portion of the training set to be involved in their operation.

2.10.4 Practical considerations

The following discussion assumes that a neural solution had already been chosen. It will be centred on the most common algorithm, multi-layer perceptron neural networks and backpropagation learning laws, although a substantial part of it is applicable to other systems.

The first main problem regards the selection of the type of network to be applied. The characteristics of the data provides an indication as to what type of network is best suited to the problem. Also the knowledge of the subject will contribute greatly to this decision, as the objectives will determine whether to use supervised or unsupervised learning. In some applications the objective of the network is to reproduce a relation between input and output states. This will impose a supervised learning strategy. An intensive application suggests the use of the backpropagation algorithm. Also the tools available could determine a preference as to one of the paradigms.

Once the paradigm has been selected than several other parameters have to be taken into account. A network topology needs to be defined. particularly with regard to the number of hidden layers and their size. All practical considerations on cost. manageability, storage, implementation, etc.. will point to the selection of the smallest network provided the network is capable of performing the required task. Despite this. one must consider that small networks can not learn to an acceptable accuracy. A large network would certainly be easier to teach and is capable of learning more patterns. although a large training set could be required to define all that a complex network can learn. A pragmatic approach to this problem could consist of the selection of a small network which will be increased until a network that ensures an

adequate accuracy is obtained. This approach can be particularly worthwhile if speed and cost are important factors. If accuracy is important then an exhaustive search of the best solution is suggested.

Another important disadvantage is that a large network could learn too much. Not only the main and relevant characteristics of the learning set, but also particularities of the training set, hence becoming an associative memory with little generalisation capabilities. A critical goal during training is to find a network large enough to learn the application but small enough to give adequate generalisation [54].

The number of layers to use is another parameter that has to be defined. Although the limitations of some configurations are known, there is no clear way of defining 'a priori' the optimal number of layers. Lippman [99] proposed a gradation of capabilities of classification for backpropagation neural networks, which has since been proven not to be true [47] [90] . One of the advantages of neural networks is that they can still be used when the number of 'data clusters' are unknown or ill defined. Also, there are theorems that show that 2 layer backpropagation networks with an infinite number of neurons are universal approximations[62] [75], but these are of little practical use as the number of hidden nodes requires could be very large.

A few other choices need to be made. As an extremisation algorithm it will start from a random position, which is determined by the weights of the training set. The only constraint is that they are non zero values, otherwise the system will be unable to evolve, due to the multiplicative nature of the equations involved. The main rule is to initialise weights in the range [-1,1],

although some authors argue that training will be faster if the range is reduced to [-0.3, 0.3] [37].

In addition to the above set of decisions, there are a few coefficients that need to be fixed. For these there is no rule known (except the range [0,1]). An example is the learning rate and the momentum coefficient. The values for the learning rate are usually in the range [0.25, 0.75] [31], although some authors had successfully used values outside this range [37]. A momentum term needs also to be specified. This can be adjusted during the learning phase, initially using a large value to allow the system to approach the minima neighbourhood quickly and then using a smaller value to allow for fine tuning. A similar procedure could be implemented at later stages to force the solution to binarize.[119]

The next factor to define is the training set. Although a full training set is desirable. and if learnt will produce a robust network, it is not always possible to obtain. Nonetheless the definition of the learning set is one of the key factors for the success of the application [55]. The training set should be as large as possible and balanced over the whole range of values. When it is not possible to obtain a full training set then a sometimes almost blind. search could be used.

Finally it must be decided what should be taught and for how long. Some neural networks reach a stable state after a number of iterations. Others just approach a solution as for continuous functions. and a decision must be made as when to stop training. In the case of classification it is considered successful when 100% of the patterns are correctly detected.

A common technique is to train until the network gives 90 to 95% correct responses, or it reaches a defined neighbourhood of the intended goal. This neighbourhood is usually defined by the error function used. In pattern recognition applications a similar criteria cannot be always accepted as a sufficient condition for success, as it could be highly affected by the type of patterns unlearnt. The whole process will be defective if the most common pattern to be presented was unlearnt.

The power of any pattern recognition technique resides in its ability to deal with noise or distortion. A high recall factor does not necessarily mean that a robust solution was obtained.

Artificial neural networks are capable of performing complex decisions. Yet, their development often consists of extensive work, requiring several training sessions until an acceptable solution is obtained. Although an apparently straightforward tool, neural networks still leave a substantial part of the system development to the experience of the user. As Philip Wasserman [179] states

> "(...) unlike other engineering discipline, there is no complete, rigorous body of science to support design decisions. Nevertheless, by a combination of heuristics, experience and whatever science is available, workable neural systems are being produced and applied with good effect" [P. Wasserman, op.cit., page 242].

2.11 Conclusion

Edge detection is one initial step in many image processing applications. Through this chapter proposed edge detection algorithms have been described, followed by a description of comparisons performed between them. Several artificial neural networks paradigms have been described and issues related to the development of neural applications analysed.

In the next chapter a new solution to edge detection is presented. The solution suggested is based on neural network arbitration between edge maps produced by different edge detectors. as they produce different edge maps. The mathematical results which show that such an approach is feasible will be stated. as will the formal mathematical description of the method used.

3 A NOVEL APPROACH TO EDGE DETECTION

"Dictor sapienti sat est"[4]
Plautus

3.1 Introduction

In the last chapter different approaches to edge detection were described as were several Neural Network paradigms. Within the current chapter a novel approach is suggested to the problem of edge detection and correct edge location identification. This being the application of multi layer perceptrons to the arbitration of edges obtained from different edge detection techniques. The suggested solution will be discussed and some assumptions or practical hypothesis that could arise described. This chapter is included to show that the approach being suggested is feasible and is soluble by a neural network, for the cases that are presented in the following chapters. The mathematical theorems that support the work and show that such an architecture is feasible are also described within this chapter. The mapping problem and one of neural network implementation is highlighted. Finally the back propagation training algorithm for the multi layer perceptron is presented in detail.

[4] What's been said is enough for anyone with sense

3.2 Neural Networks and Pattern Recognition

Recognition problems usually take into account a set of measurements or features and consider them as a point in a n dimensional feature space. Various algorithms can then be applied to divide this feature space into compartments, which will be used in the classification. A significant problem is the boundary placement in the feature space so as to allow for the decision regions to be correctly formed. The underlying assumption is that points that belong to the same class tend to cluster. The complexity and extension of the domain do not allow this regions to be formed by a simple mapping, thus requiring the use of more complex techniques to perform the boundary placement. Neural Networks are the selected tool, as they are able to overcome the fuzziness and misdefinition of the mapping to be performed.

Conceptually, the problem is clear to define. It is required to identify an operator, which maps a relation between the domain and the objective sets. The domain is defined as subsets of the edge maps, extracted and operated upon to produce a value identifying the existence or non-existence of an edge. These subsets are defined as a vector resulting from successively aligning the rows of square window of the image, extracted from each of the edge maps used. The output is assumed as a binary value stating the fact that the central points is or is not an edge. In practice the problem is not so clearly defined, due to a lack of clarity in the definition of an edge and due to the fact that the relation can not be defined as a map due to the nature of the image.

3.3 Arbitration

As stated earlier the objective of edge detection is to produce from an image an associated image where all the edges and only the edges are marked. For the purpose of the following text let us assume that for every image A the associated edge map exists, which itself is an image, and is designated by E_A.

Several approaches, described earlier, have been developed to define a function such that

$$E_A = \xi(A) \tag{29}$$

which maps the relation ξ between the image A and their corresponding edge map E_A.
These different approaches will be referred to as $\varphi_A, \varphi_B, ..., \varphi_n$, which , when applied to the image I, will produce images containing the edge maps $\Phi_A, \Phi_B, ..., \Phi_n$ respectively, such that $\Phi_A = \Phi_A^I = \varphi_A(I)$.

The objective is to search for an operator ω_i, which instead of accepting the image I as an argument, accepts an ordered set of processed images $\{\Phi_A^I, \Phi_B^I, ..., \Phi_N^I\}$ as argument, such that

$$\begin{aligned} \Omega_i &= \omega_i(\Phi_A^I, \Phi_B^I, ..., \Phi_N^I) \\ &= \omega_i(\varphi_A(I), \varphi_B(I), ...\varphi_N(I)) \end{aligned} \tag{30}$$

will produce a better approximation of E_A than each of the arguments $\Phi_A^I, \Phi_B^I, ..., \Phi_N^I$ by themselves.

The operator ω_i should not process the whole image in a global way, as none of the functions φ_i process the image in a global way. As referred to in chapter two some of the operators are defined as a convolution between the image and a local operator or mask. Lets designate the raster scan operator as $\vartheta(\ )$. This operator just repeats over the whole image, row by row, the function in the

argument. Using this operator, some of the operators referred to in chapter two could be defined as

$$\varphi_i = \vartheta(M(x,y)) \tag{31}$$

where M(x,y) stands for a mask or local operator, which operates over a limited area of the image

The same strategy could be applied in the definition of the function ω_i, as $\omega_i \equiv \vartheta(\chi_i)$. Rewriting equation (30) as:

$$\Omega = \vartheta(\chi_i(\Phi_{A,x,y}, \Phi_{B,x,y}, ..., \Phi_{N,x,y},)) \tag{32}$$

where χ_i is the function looked for, and which is able to perform the arbitration between corresponding small areas in edge detection maps. This form will also allow, in some cases, to imbed the arbitration process in the system without the necessity of generating intermediate pictures. Or, in other words, instead of

$$\Omega = \vartheta(\chi_i(\vartheta(M_A), \vartheta(M_B), ..., \vartheta(M_N))) \tag{33}$$

it can be generated as

$$\Omega = \vartheta(\chi_i(M_A, M_B, ..., M_N)) \tag{34}$$

The form (34), if the functions M_A, M_B, ..., M_N allow, is able to permit a more efficient implementation of the operator and is preferable to the form (33) and will be researched. This is due to several practical reasons. Firstly to allow the use of operators which do not assume the form (31) (e.g.. Deriche, see [33] [34]). Secondly in the development phase, several networks will be tested over the same tuple of images. This allows for rapid development, avoiding the need to repetitively apply the same operator on the same image.

The arbitration operator χ_i accepts as input the vectors constructed from corresponding windows of the image. Assuming a non commutative operator, and thus ordered arguments, the domain will be the reunion of the contra domains of each of the partial operators. This fact does not limit the scope of the operator being defined. Assuming the arbitration process works over a square window of w x w pixels, then the domain space for the operator χ_i will have the dimension $D=nw^2$. Windows will be restricted to odd values (w=3,5,7,...), so that they can be centred on one pixel of the image array.

The function χ_i will accept a limited range of values, since it will operate over quantified images. The edge maps are restricted to the usual range {0...255} by normalisation. This normalisation is performed to maintain the results within the range of values common in the image processing hardware. This normalisation is also performed as it limits the input to the arbitration system.

The neural network addresses the problem of the approximate implementation from a bounded subset D of a n dimensional Euclidean space to a bounded subset f(D) of a m dimensional Euclidean space, through the use of input/output samples (ip_1, op_1), ($ip2$, op_2), ..., (ip_k, op_k), ... such that $op_k=f(ip_k)$. The existence of a simpler network was stated by R Hecht- Nielsen as the 'Kolmogorov Mapping Neural Network Existence Theorem' (op.cit, pp 122):

> "Given any continuous function $f: [0, 1]^n \rightarrow \Re^m, f(x) = y$, f can be implemented exactly by a 3 layer feed forward neural network having n fanout processing elements in the first layer, (2n+1) elements in the middle layer, and m processing elements in the top layer"

The fact that the theorem is stated for an input defined in the unitary cube does not induce any other problem than the need for normalisation of the input

vectors. As the arbitration domain is by construction limited, it will suffice to normalise it by its upper boundary (255), for the function χ_i to be under the conditions of the theorem. This function is assumed as continuous. The theorem proves the existence of such a network. Unfortunately it does not suggest how it could be implemented, as Hecht Nielsen states. Although several authors assume that Neural Networks with 2 hidden layers are universal approximators, as stated by Vera Kurkova [94], the existence of a simpler network was stated by R Hecht Nielsen (op cit., pp 133):

> "Given any $\varepsilon > 0$, and any L_2 function $f: [0,1]^n \rightarrow \Re$ there exists a three layer back propagation network that can approximate f to within ε mean square error accuracy"

L_2 is a function where each of the f's co-ordinates are square integrable on the unit cube. This functional space includes the continuous functions, and includes all discontinuous functions that are piecewise continuous on a finite number of subsets of $[0,1]^n$ (idem).

This result suggests that a neural network is able to approximate the required operator function. Even if the function is assumed as discontinuous, the number of discontinuities is limited, due to the space quantification that results from the normalisation procedures. Although stating only the existence of an approximation it is enough for the purpose of the research presented.

In many practical applications more than one hidden layer is often essential, as if a simple layer was used the number of neurons would be impractical. As Hecht Nielsen states that to obtain the above result:

"(...) provide the confidence that comes from knowing that an appropriate backpropagation architecture must exist" (op.cit, pp .133)

When using actual configurations and learning laws some limitations of the mapping capabilities of these types of networks are described [47] [90].

3.4 Structure of a multi layer perceptron

The term multi-layer perceptron is used where one or more hidden layers exists. The basic example is presented in figure 14. This presents a two layer perceptron network which contains two layers of weighted connections. between the input and the output. Each slab of neurons is defined by a vector which defines the geometry of the network. In this case the vectors are defined as I(i), H(h) and O(o) for the three slabs going from the input (I) to the output (O). The weights are defined by the matrices W_{hi} and W_{oh}, where the indexes are indicated in reverse order. These weighted matrices have the dimensions W_{hi} [H.I] and W_{oh}[O.H] respectively.

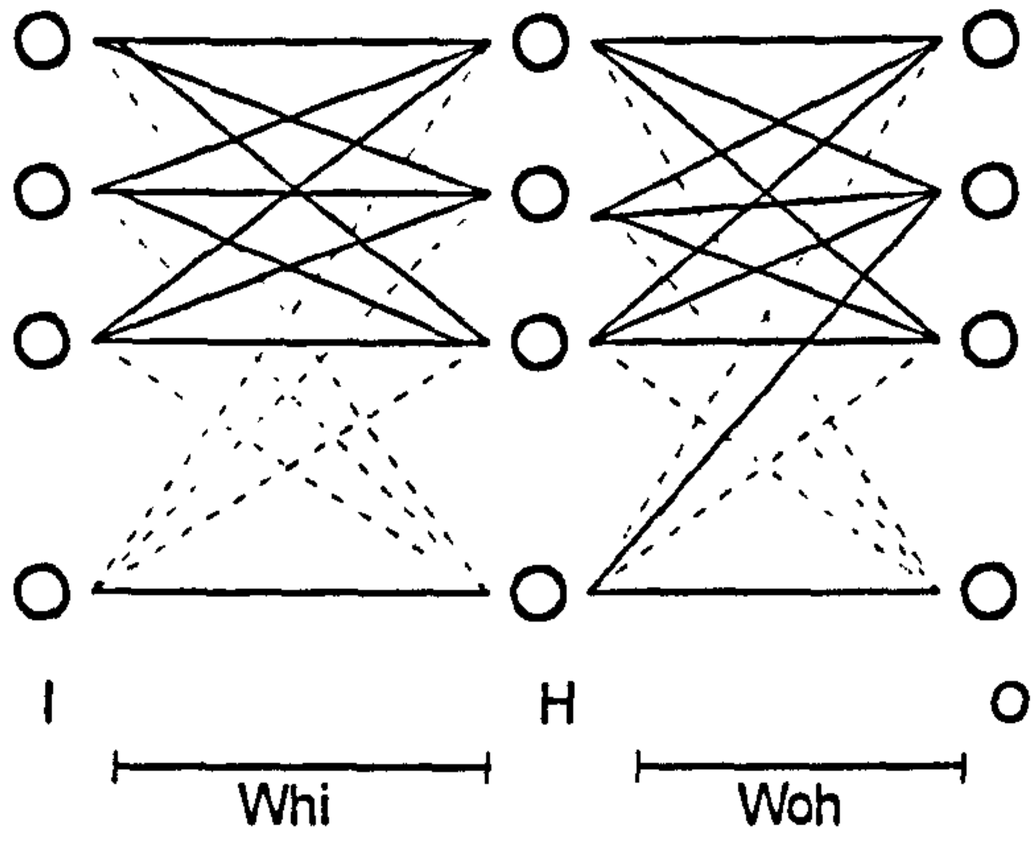

Figure 14: Perceptron structure

Each neuron contains a value, designated as its activation function. The dimension of the input vector I is equal to the dimension of the input pattern (plus 1 if a bias is used). Similarly, the dimension of the output vector O is the dimension of the output pattern. If a bias neuron is used, the input and hidden layer will have an extra term, being in this case the coupling matrices of dimensions W_{hi} [H,I+1] and W_{oh}[O,H+1] respectively. This term will allow a non null output for a null input. This extra neuron is assumed to always have an activation value of +1 and thus will affect all neurons in the following layer only by the value of the weighted connection.

3.5 Forward Step

In the case of the forward pass learning phase a number of patterns are presented to the network. The nodes in the input layer only receive the patterns that are presented to the network and act as a buffer without performing any

processing function. Each node of the following layer is a weighted sum of the node values of the previous layers operated on by a function F( ). Assuming that a pattern P[i], with dimension I, is presented to the network, the value of each node in the hidden layer is defined by

$$\mathbf{H}(h) = F^h(\sum_{i=0}^{I-1} W_{hi}\mathbf{I}_i + W_{hI}) = F^h(\sum_{i=0}^{I-1} W_{hi}\mathbf{I}_i^*) \tag{35}$$

where $\mathbf{I}_i^*$ is a completed input vector, with I[I]=+1.

The calculations for the following layers are identical and the same rule is applied

$$\mathbf{O}_o = F^o(\sum_{h=0}^{H-1} W_{oh}\mathbf{H}_h + W_{oH}) = F^o(\sum_{h=0}^{I} W_{oh}\mathbf{H}_h^*) \tag{36}$$

In the remaining text a bias term is denoted by *.

The function F() is designated as the node threshold or transfer function. Usually the same transfer function is used for both layers, with $F^h \equiv F^o$.This function must be non decreasing and differentiable. The sigmoid function defined in (37) is the most common choice

$$S = \frac{1}{1+e^{-\gamma x}} \tag{37}$$

Where γ is the slope of the function, usually set to one. The calculations of the following layers are identical and the same rule is applied.

3.6 Learning Objective

Consider a given multi layer perceptron and a respective learning set $\bar{L}_p = \left(\bar{I}_p, \bar{O}_p\right)$. For each realisation of the neural network weights, a vector of actual outputs is produced

$$\bar{O}_a = N(w) \times \bar{I}_p \tag{38}$$

The learning objective is to solve this equation in order to find N(w), allowing the definition of a weight set W_L such that

$$\bar{O}_p = \bar{O}_a(W_L) = N(W_L) \times \bar{I}_p \tag{39}$$

As an algebraic solution is not known, an iterative solution was proposed by [145] as the minimisation of the function

$$E(w) = \tfrac{1}{2}\left\|\bar{O}_p - \bar{O}_a(w)\right\| \tag{40}$$

Among the different techniques for minimising a function of $\Re^n$, back propagation is based on a steepest descent technique.

The underlying idea of a steepest descendent technique is to move from the current position in space, in the best direction, to one with a smaller error. This learning rule was extended to the form known as the Generalised Delta Rule [145] and allowed for the development of a learning rule for multi-layer perceptrons.

3.7 Backward pass

The back propagation algorithm is based on the generalised delta rule. In the forward step, described previously, the calculations are performed starting

from the input layer, through every other layer until the output is obtained. From a feed forward pass, the error between the actual and the desired output is calculated. In the backward pass this error is propagated through the network in order to change the weights in a suitable way.

The error term δ is for a generic node given by (updating from pass k to k+1)

$$\delta_o^{k+1} = \mathbf{O}_o(1-\mathbf{O}_o)(\mathbf{O}p_o^k - \mathbf{O}_o) \tag{41}$$

and for the hidden nodes

$$\begin{aligned} \delta_h^{k+1} &= f'(\mathbf{H}_h)\sum_{h=0}^{H} w_{hi}\delta_h^k \\ &= \mathbf{H}_h(1-\mathbf{H}_h)\sum_{h=0}^{H} w_{hi}\delta_h^k \end{aligned} \tag{42}$$

The terms of the form X(1-X) are a more efficient form of the derivative f' of the sigmoidal function. which should be used as the transfer function during the learning phase.

The weight matrices are updated from the output to the hidden layer using

$$\mathbf{w}_{ho}^{k+1} = \mathbf{w}_{ho}^k + \eta\delta_o\mathbf{O}_o \tag{43}$$

and

$$\mathbf{w}_{ih}^{k+1} = \mathbf{w}_{ih}^k + \eta\delta_h\mathbf{H}_h \tag{44}$$

The learning constant η is predetermined for the system.

Some authors add a momentum factor which has been shown to increase the convergence characteristics. given by

$$m_j^k = \alpha(\Delta w_{ij}) \tag{45}$$

and is proportional to the weight change in the previous iteration. The weight update equations being written as

$$\mathbf{w}_{ho}^{k+1} = \mathbf{w}_{ho}^{k} + \eta\delta_o \mathbf{O}_o + \alpha\Delta w_{ho}^{k} \tag{46}$$

and

$$\mathbf{w}_{ih}^{k+1} = \mathbf{w}_{ih}^{k} + \eta\delta_h \mathbf{H}_h + \alpha\Delta w_{ih}^{k} \tag{47}$$

3.8 Convergence criteria

The backpropagation algorithm is an iterative process and as such it will continue indefinitely getting successively closer to the desired extreme. However, as the step is a fraction of the distance, it never reaches the extreme, but, can go as close as desired.

The parameter used to perform the minimisation is the error term, calculated at the beginning of the backward pass in the learning process. The error term used is the mean square error (calculated over the full set of p training patterns), which for a network with k output nodes is

$$Ep = \frac{1}{2p} \sum_{p,k} \left(O_{p_{pk}} - O_{a_{pk}} \right)^2 \tag{48}$$

being the coefficient $\frac{1}{2p}$ usually omitted.

However. in our case this parameter is not meaningful. as the main objective is for the patterns to be learnt and not the minimisation itself. Thus the counting of the patterns learnt is a better measure of the success of the learning phase. The best pattern recalling factor does not always correspond to the minimum average squared error. The fact that the error is decreasing, iteration after

iteration, does not imply that the number of patterns being learnt is increasing. Although a more robust solution is achieved and to some extent should be allowed. it will be meaningless to allow the learning phase to continue indefinitely. Due to these reasons, the number of correctly recalled patterns is also monitored. This value is more significant as a monitoring criteria, since it measures more accurately the objectives of the learning phase.

3.9 Conclusions

In this chapter the application of multi-layer perceptrons to the arbitration problem was discussed and some assumptions or practical hypothesis that could arise described. Next, the mathematical theorems which support the work being carried out were presented. Finally the back propagation training algorithm for the multi layer perceptron was presented in detail.

The next chapter describes the implementation upon a parallel processing platform of the algorithm presented in this chapter. The main parallel processing paradigms are presented first, followed by several experiments that were performed. Finally comparisons of different implementations are performed and conclusions drawn as to the best implementation solution.

4 SYSTEM IMPLEMENTATION

Parallelism is the norm; Purely sequential problem solving is the anomalous restriction (Carriero [20], page 1)

4.1 Introduction

In the previous chapter the approach being researched was presented, as were mathematical results which show that such an approach is feasible. The neural network paradigm chosen was presented in detail, together with the equations used for the algorithm being developed.

In the current chapter the implementation of the system upon a parallel processing platform will be described. The system described consists of a set of transputers, where the computational load is spread between them. However, the algorithm spreading has an adverse effect, as an extra load is introduced due to the necessary communication management. This requires algorithm efficiency monitoring throughout the development steps of the parallel system.

This chapter has the following structure: Initially an optimal solution for the neural network implementation upon the transputer network is investigated and evaluated. In section 2 the transputer and some of the basic paradigms for

parallel processing are described. From these basic paradigms the most promising is chosen for the implementation of the neural network. The tests performed to assess its validity and advantages are described in section 3. In section 4 the results obtained are discussed. A second implementation paradigm, which leads to the final system. is next to be presented and discussed.

4.2 Parallel processing - transputer approach

The most common computing architecture used is sequential and is usually referred to by the acronym SISD. for Single Instruction Single Data. Although sequential computers are acquiring higher performance daily, parallel processing architecture can offer higher performance when implementing inherently parallel algorithms.

Although a general parallel processing architecture classification does not exist. the architecture grouping due to MJ Flynn [40] is usually cited in the literature. Flynn classifies computer architecture in four classes. One of them. is the traditional sequential architecture. the SISD machines. Flynn's classification resides in the parallel nature of the machine. The second classification is designated as the SIMD architecture (Single Instruction Multiple Data). The third as the MISD (Multiple Instruction Single Data) architecture. For this architecture different processes can be performed at the same time. on the same data. The final architecture is the MIMD (Multiple Instruction Multiple Data) architecture. This consists a set of processors executing different programs. on different data. with the possibility of changing information between them. This is the most powerful. general and versatile case of Flynn's classes. It includes all the other classes as it is possible to implement the others classes using such a system. An example of

such a machine is a transputer[5] network. A transputer can act as a conventional computer. or several transputers can be connected together to form a truly parallel system allowing for a wide choice of algorithm implementation.

4.2.1 Transputers

The transputer is a single-chip processor with is own central processing unit and memory. It also has the ability to exchange data with other transputers or devices. It allows the development of a truly parallel network of processing elements. which can be configured to suit the purposes in an optimal way. Each transputer has four communication channels and includes internal hardware support for parallelism. This allows a variety of parallel architecture's to be built and a wide range of options for program design.

The allowable architecture is limited by the number of physical connections allowed. Some examples are presented in Figure 15. For the work reported in this thesis a further limitation exists, as the network is mounted on a TMB16[6] transputer board. For this board one link of the first transputer is connected to a switch controller. The use of this type of board allows for versatile hardware configuration. Although it originates an additional constraint on the network topology it allows for alterations to the network configuration. Thus several architecture can be implemented. This is done using the Network Configuration Software (NCS™) provided by the manufacturer. The network used in this work is composed of a variable number of T8 transputers .as most of the algorithms work on real values. However. a system could be implemented using a T4 network. by suitably changing the range of values so

[5] Transtec
[6] Inmos TM

as to allow for integer arithmetic.

4.2.1.1 Transputer applications development

Throughout this work the transputers were programmed in Occam[7], which is a language specially designed for them. There are other languages available, such as *3L Pascal* or *3L Parallel C*, however the Occam structure is very efficient to run on a transputer network.

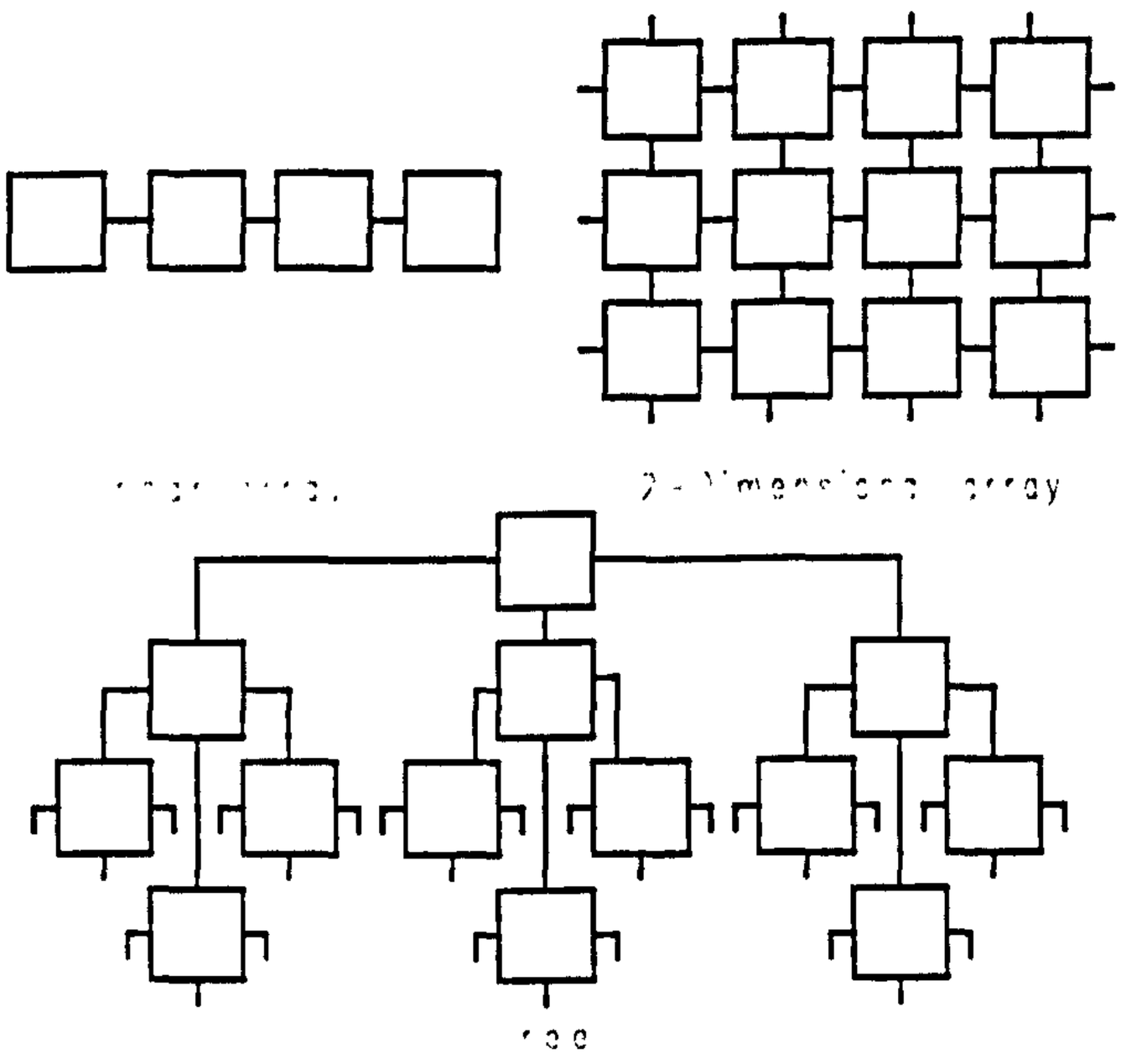

Figure 15 : Some basic transputer architectures

4.2.1.2 User Interface

The transputers are initialised through a C program, which boots and loads the executable code. This program was downloaded from the host computer onto the TMB16 board transputers, where it is used to perform the main calculations while the host PC provides the graphical interface, as the

[7] Inmos TM

transputer board does not provide any graphical capabilities. This procedure also allows the use of the storage and retrieval functions of the C compiler. Acquisition, display, storage and retrieval of images, are done through this interface, as occasionally is some auxiliary processing. The storage and retrieval of the neural networks data is also done through the C program. Also the image display was done through the C interface, which can access the trident chip, allowing for the display of 64 grey levels[8], or alternatively, the 16 grey levels provided by the VGA display. The display follows the access procedures described by Stevens [160] All the transputer code described throughout this thesis has a companion C program for the user interface.

Communications are done using a set of procedures listed in the INMOS Data sheet for the B008 board [171]. These, along with the procedures to directly access a frame grabber [135] are the only library routines used through the work.

4.2.2 Concurrency vs. Parallelism

Concurrency and Parallelism, although similar concepts, should be defined precisely. When refering to programs running in parallel we mean that the programs run on different hardware, and thus they do not interfere with each other except when some communication is performed.

Alternatively a processor can execute simultaneously any number of processes. Thus processes share the processor time. This is referred to as 'time slicing', since the processor consecutively allocates a slice of time to each of the processes. A concurrent [111] program is defined as a program which has

[8] Although it allows 256 colours, each of the RGB components allows 64 intensities, and thus the number of grey levels possible to display are 64.

all the characteristics for parallel processing, but is processed on an alternate basis, giving the CPUs attention to different processes successively. This time sharing is done by a built in scheduler. The scheduler operates in such a way that inactive processes do not consume any processor time. In general a concurrent program runs slower then a corresponding sequential version due to the scheduler's overhead. Thus it is best to avoid the use of concurrent processes in a program, except when strictly necessary. This is also true when related to the mapping of several processes on a transputer network. The preferred procedure is to map different processes over different transputers, which truly work in parallel. However, to optimise the allocation of resources, the use of concurrent processes could be good practice, since it will maintain the versatility of the software while minimising the hardware costs involved.

4.2.3 Communication model

Another important notion when implementing algorithms on transputers, is the communication model used. Transputers communicate using channels, which are associated to a protocol. Communications in Occam are synchronised. A program will wait during a communication process until the data is sent or received by the other communicating process. Communications act as time regulators and can be detrimental to the performance of the program, since they can originate dead times due to careless disposition of communications. It is important to accurately position the communications between parallel / concurrent processes so as to obtain high program efficiency.

It is also important that processes do not stop the program whilst waiting to receive (or send) values simultaneously. This is known as 'dead lock'. In the algorithms being implemented, this situation will not exist since the

communication sequence is predictable and adheres to a strict sequence.

Communications are also time consuming and thus should be kept to a minimum. Also, the passing of values should be done in blocks. This will allow for the efficient use of the transputer communications protocol, thus reducing the data transfer time.

4.2.4 The configuration of a transputer network .

The transputer programs were developed using the Occam Toolkit and Transputer Development System (TDS)[9]. Occam allows for the transputer network to be configured at the hardware or software level, thus allowing for the definition of virtual nodes. This allows for a more versatile program structure. since independent programs can be allocated to the same transputer. Also, the whole system could be adapted to changes in the transputer network without changing the software or the logical network used.

4.3 Fundamental Parallel Paradigms

4.3.1 Introduction

When programming a network of transputers, a large degree of freedom exists in the implementation of algorithms. In a sequential computer the permutations of single independent processes does not. in the majority of cases affect the performance. This is not so for a parallel processing system. The best performance will be achieved by the maximum use of the processors.

[9] Transputer Development System. Inmos Inc.

thus process allocation will play a crucial role in the implementation efficiency. Another key fact is that, if the number of elements in the network is incremented then the total processing time is decreased. at the same time the cost will increase. There will always exist an optimum value for the ratio of speed / cost, which will be more application dependent than algorithm dependent.

The following subsections will briefly refer to some common paradigms for distributing processes over several processors. For each approach, the best suited type of algorithms will be included.

4.3.1.1 Task or data parallelism

An early classification of computer programs divided them into two categories, (i) the ones which carried out many operations on a very small amount of data and (ii) the ones that do not carry out many operations but process a very large amount of data. However, as with all classifications there are exceptions. This classification will allow for the investigation of the two basic paradigms. The core of the question is which is more efficient. to split, the data or the task?

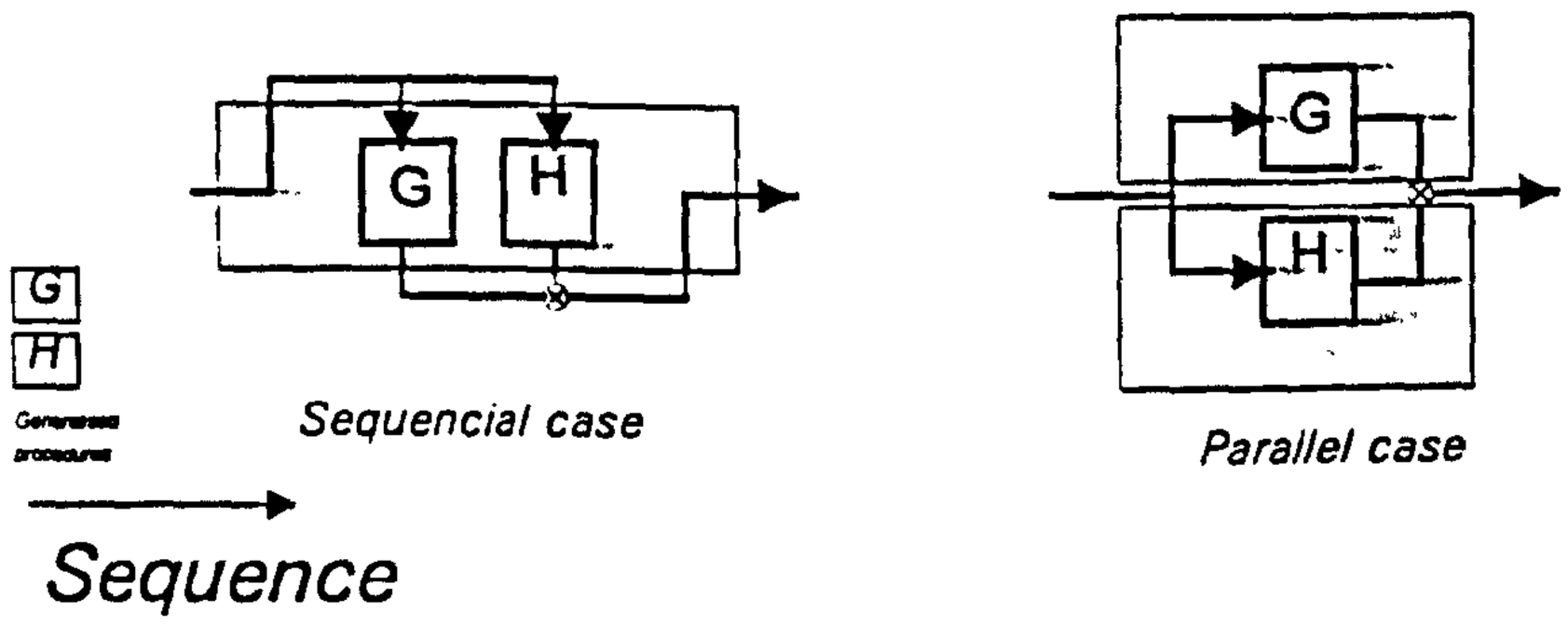

Figure 16: Task Parallelism

In the first case, 'computation bigger then data' type of algorithms, results are usually obtained by the combination of several independent processes, applied to the same data, which in many cases can be performed independently.

The sequential solution is to perform processes one after the other without any pre-defined order (see Figure 16). The natural parallel solution is to divide the algorithm into the different independent blocks, and to allocate each block to a different processor. Then join the results at the end. The time reduction is dependent on the number of parallel processes and their length. Communications have not yet been considered .

The second case, 'data bigger then computations', a small number of calculations are successively applied to a large number of data sets. A typical case is a convolution operator, a category in which a large number of edge detection filters are included. Taking as an example the Roberts edge operator (the mean square version). Then two multiplication, three additions / subtraction and a square root are applied to sets of 65 thousand points. (assuming a 256^2 image). The obvious approach is to split the data over several processors executing the same algorithm and collect the results in the required order (see Figure 17). Data parallelism is a natural approach to parallel processing for many image processing algorithms. It is used when the same operation is performed successively over independent sets of data. In the case of a convolution filter the data set is split over N_{trans} transputers, allowing a reduction in the processing time T_{seq} to

$$T_{seq}/N_{trans}-T_{coord} \tag{49}$$

where T_{coord} is the time consumed in the division of the image over the several

processors.

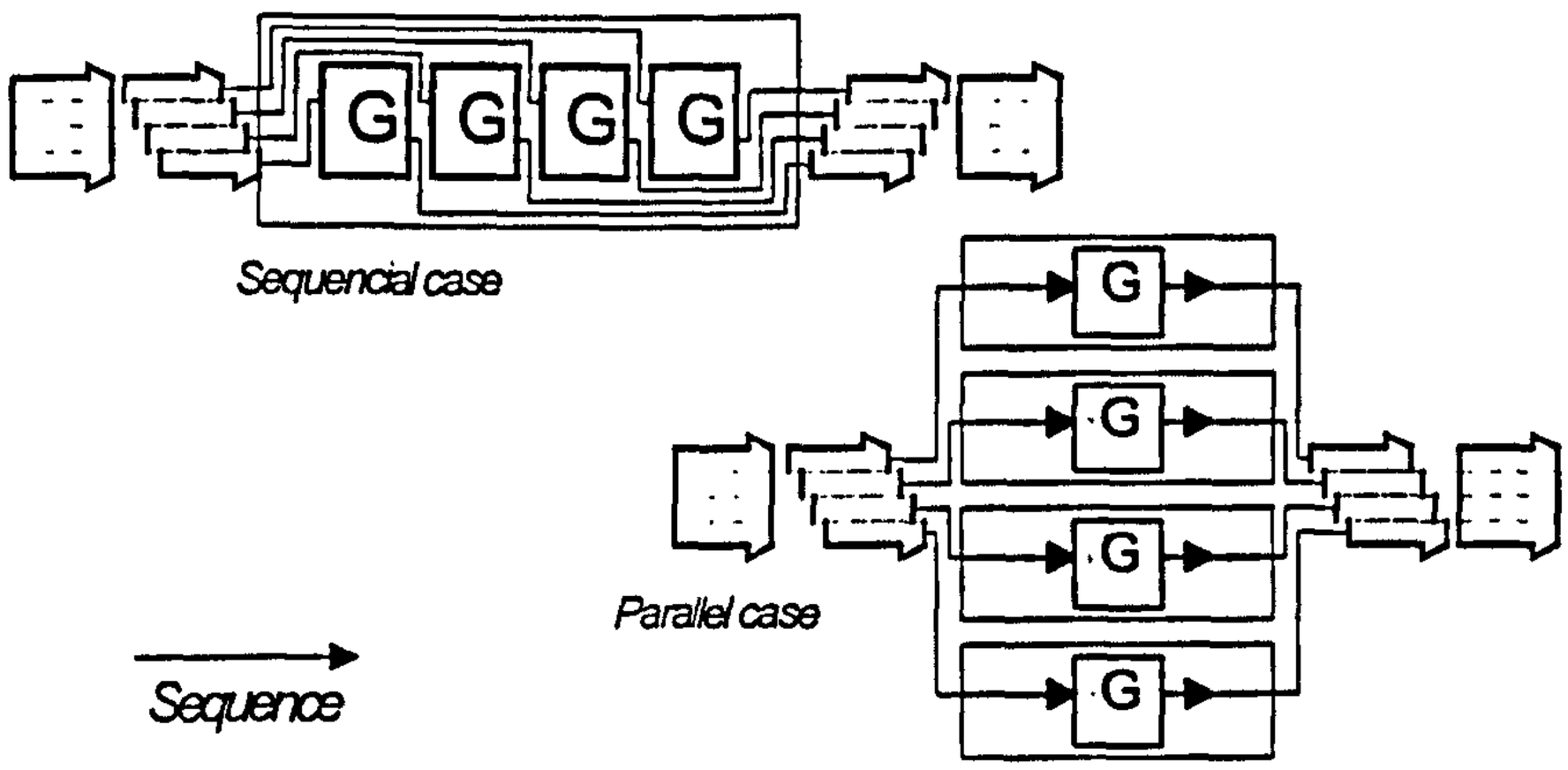

Figure 17:data parallelism

These two topologies are designated as MISD and SIMD architecture's. respectively.

4.3.1.2 A process farmer

Frequently an intermediate situation between these two cases exist. Some condition. sparse over the data, could in a random way increase the computing time of one of the processes. and thus unbalance the system. The solution is to spread the algorithm over an adequate number of several processors and feed the data to the processors. as they became free. This correspond to a combination of the previous cases.

4.3.1.3 A pipeline

An important group of algorithms. such as recursive algorithms. are not suited to any of the implementation cases referred to so far. Some algorithms. performed over an extensive series of values. must be sequentially performed

due to the fact that at step n+1, information is required which has been calculated during step n. As soon as this information is known step n+1 could start, even if all of step n has not been completed. Thus partially overlapping the current sequence with is predecessor can be carried out. An example is schematically represented in Figure 18. The temporal sequence of the different processes is shown processing a sequence of data items. In this case, three independent sub-processes are concatenated. After the initial data has been through the system the data throughput will be the execution time of the slowest sub-process.

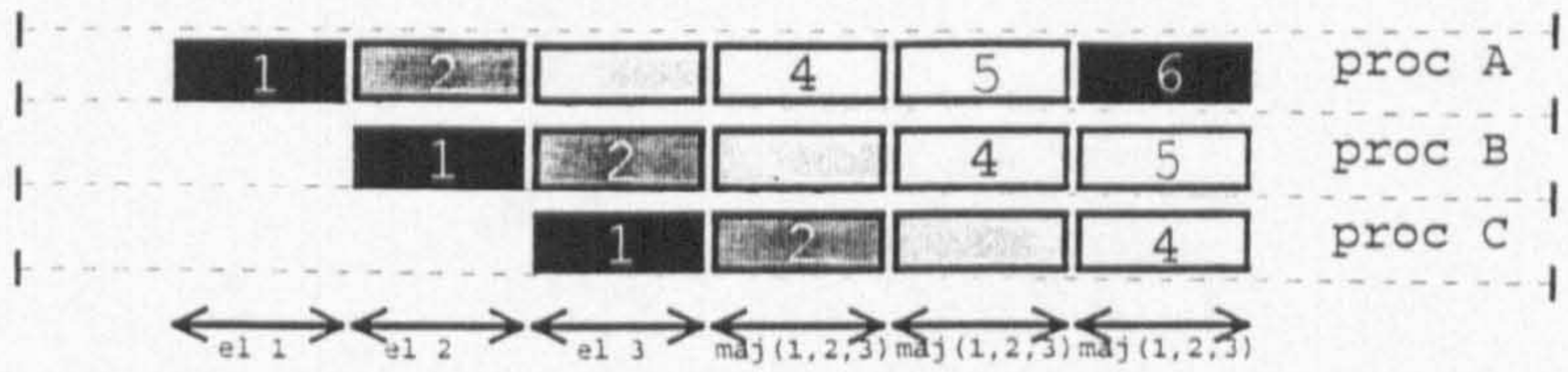

Figure 18:Time sequence of a process with 3 steps in a three processor pipeline

4.3.2 Efficiency and Granularity

The advantages of the distribution of a program over several processors is a compromise between the reduction in execution time, due to the slicing of the algorithm, and the increase in communication time. This factor is usual referred to as granularity. The minimum time an algorithm can be reduced to by a parallel processing system is bounded by the communications time.

Quite often the advantages of the use of one of the paradigms are not always clear and the difference between a parallel and a sequential process distributed by several processors could be so small as the change of a PAR by a SEQ construct in one of the programs. As an example, consider the pipeline shown in Figure 19. The first processor (A) acts as a controlling system and buffers

the communications with the host computer. The core task in each of the processors is irrelevant for the discussion. and is assumed to run without interference of the communications sequence. Processor A, sends values to processor B and receives the results from processor C. The program running in processor A has four actions :

a1 - Receive data from host
a2 - Send data to B
a3 - Receive results from processor C
a4 - Send results to host

Process B and C, receive values from the previous processor, processes them and send them on. This structure is advantageous where values are presented sequentially, as seen in Figure 18. In this situation when process C is processing the nth data. process B is already processing the (n+1)th data point. and so on. Due to the synchronism in communications, if sub - processes a2 and a3 are performed sequentially then the system will work sequentially. Process A. will stop in sub-process a3, waiting for the results. Process B will be waiting for process A to send new values, but process A is stopped in sub-process a3 waiting to receive values from process C. Only after process C had finished will process A return to process a2 and is then able to send a new set of values to process B. At this time process C is waiting for the results of process B and thus, the overlapping processing time is null. The advantages of the pipeline. are only obtained if processes a2 and a3 are performed in parallel. so that process B can feed at the same time as Process C is unloaded. and without any interference.

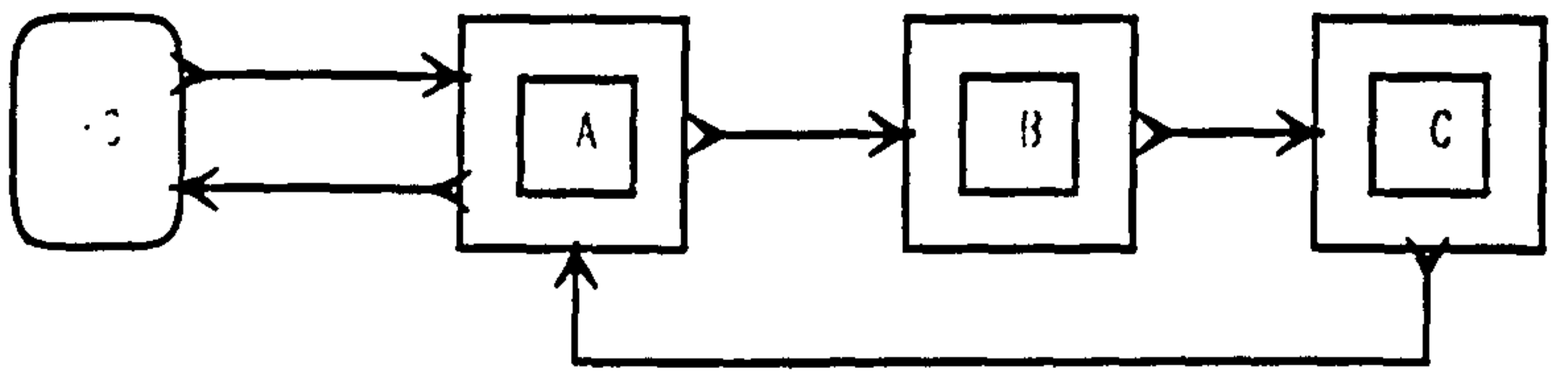

Figure 19:Basic pipeline structures

It is important to check during developing the architecture that the increase in speed obtained by the different allocations maximises the use of the available resources.

The relationship between the duration and the number of processors is not linear, due to the communication term. In addition, there are other factors that can affect the optimisation time of parallel programs. For this reason it is strongly advisable to access continuously the improvements achieved by the parallelisation process in order to maximise the system performance.

Through this work the performance of the parallel implementation will be compared to the sequential solution. This comparison is carried out upon the same implementation so that other factors do not interfere whilst the comparisons are being made. Also, the intermediate allocations will be tested. This procedure is useful so as to easily locate critical length paths of the system performance

4.4 Neural Network Implementation

4.4.1 Pipeline implementation of a Multi-Layer Perceptron

The input patterns to the neural network are vectors extracted from different edge maps concatenated. During the learning process the extraction and concatenation process are implemented separately. However, it will be impractical for the application itself to implement them separately as it is time consuming and the amount of intermediate data generated increases unnecessarily. The first process (process A in Figure 19) receives the images from the host computer and generates the vectors that become the input values to the network. To this process was initially allocated one physical transputer (M) as in Figure 20. The process sends, concurrently, the vector for the next process and waits for the results to be presented by the last process. Another transputer was initially allocated to each of the other two processes (processes H). These two processes are identical so as to allow for the addition of layers by interposition of the same program. The communications are performed in a ring.

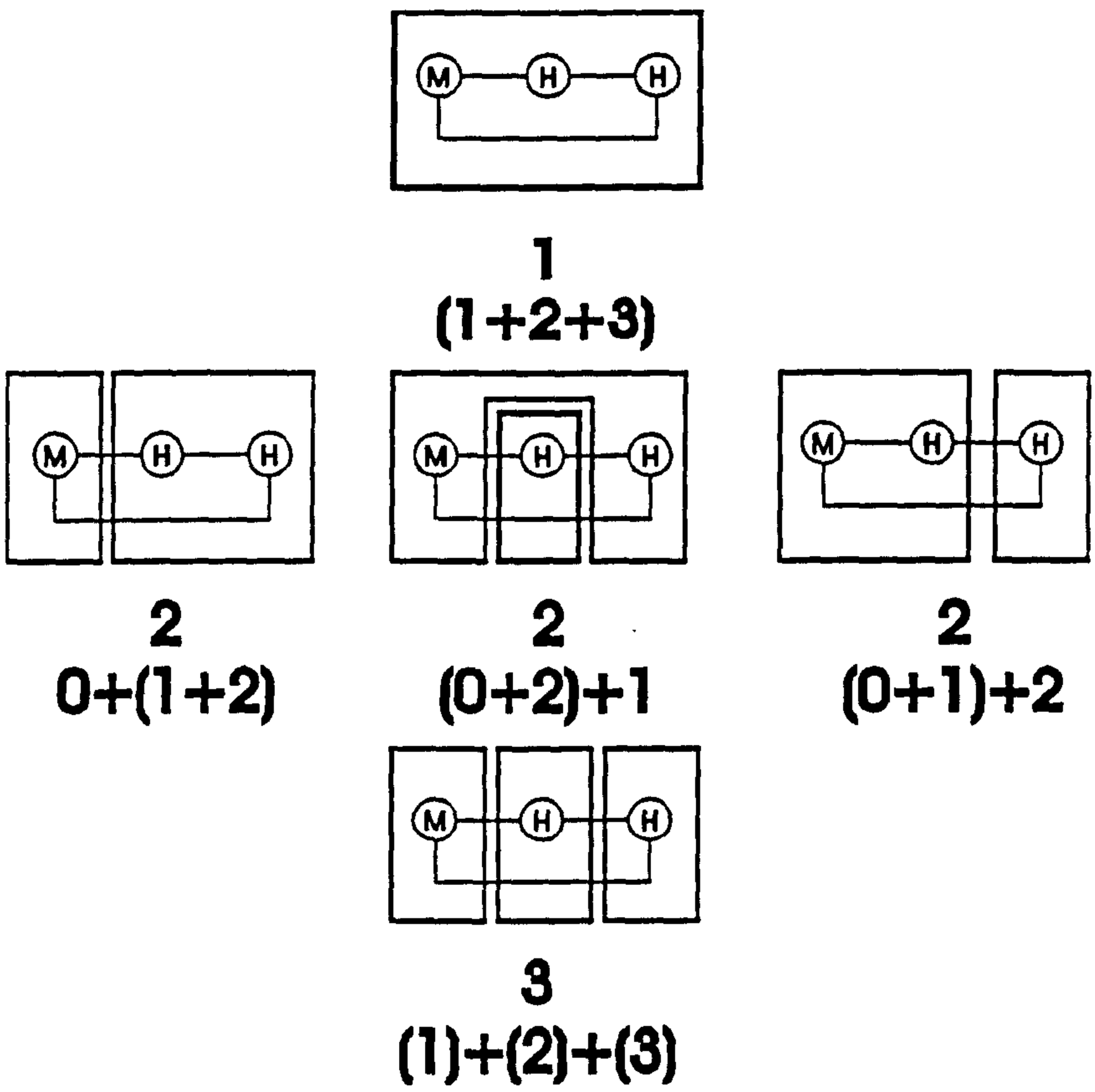

Figure 20:Process allocation on 1, 2 and 3 transputers

Due to the unbalanced size of the networks used, the processing time due to one of the layers could override the other processes execution times. In this case some of the remaining transputers will be superfluous and could be allocated to different jobs. The different allocations tested, are represented in Figure 20 and computing times obtained are shown in TABLE I. These values refer to the time taken, in seconds, to compute 9 lines for two images of a diverse nature (A and B).

TABLE I
Execution Times on a Pipeline implementation

Configuration		Network Size							
		18.2.1		18.20.1		50.25.1		50.50.1	
Transputers	Processes	A	B	A	B	A	B	A	B
1	1	15	10	116	76	288	192	566	339
1	3	17	12	119	85	291	176	572	345
2	0+12	11	8	91	65	223	135	444	267
2	01+2	15	9	113	79	285	170	447	270
2	02+1	11	9	92	67	225	137	561	335
3	3	9	6	87	61	218	130	436	259

Time ratios are shown in Figure 21. They are calculated as the time taken by the algorithm to run on the number of transputers divided by the time on a single transputer. Process allocation follows the sequence in TABLE I.

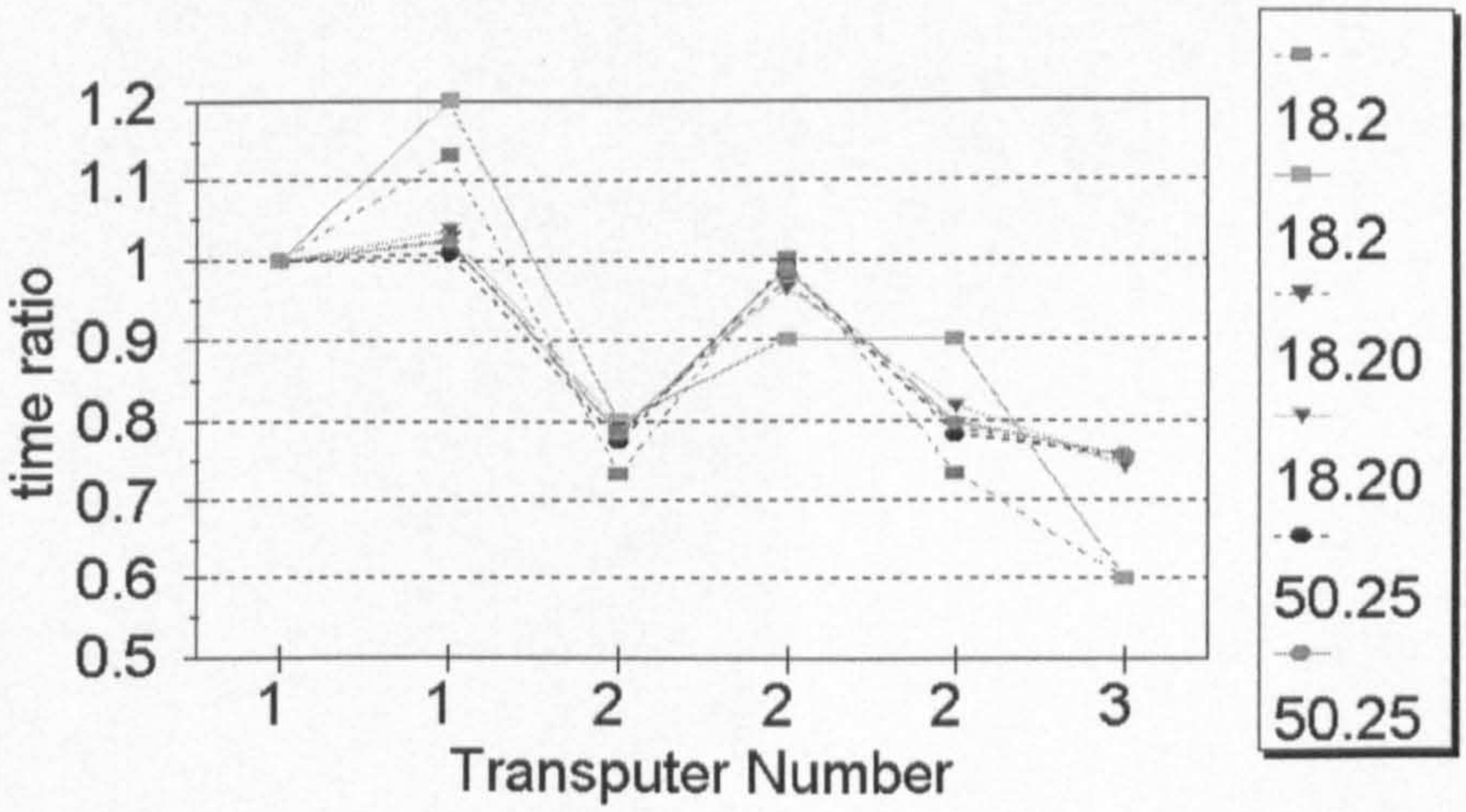

Figure 21:Allocation time ratios for the pipeline implementation

From the graph it can be seen that the allocation over three transputers is the most efficient choice. Another interesting point is that although ratios of 1/3 could be expected they were not obtained. This is due to the communication time used to propagate values between processes. Optimisation of the

processing time for layers is next to be considered.

4.4.2 Layer processing

With the exception of the master process (M) each process implements a layer of the network, as represented in Figure 20. Each H process comprises four main actions:

1- Receive vector
2- Multiply the coupling matrix by the vector
3- Calculate sigmoidal of each element
4- Send vector to the next layer

From these the two most computationally intensive actions, that can be split over more then one transputer, are the matrix multiplication and the calculation of the sigmoidal function for each of the elements. These are the areas where distribution could reduce the processing time.

For this purposes, the transputer network was increased to the network shown in Figure 22. The mapping of the neural network on the transputer array is again done in layers and implemented as a pipeline.

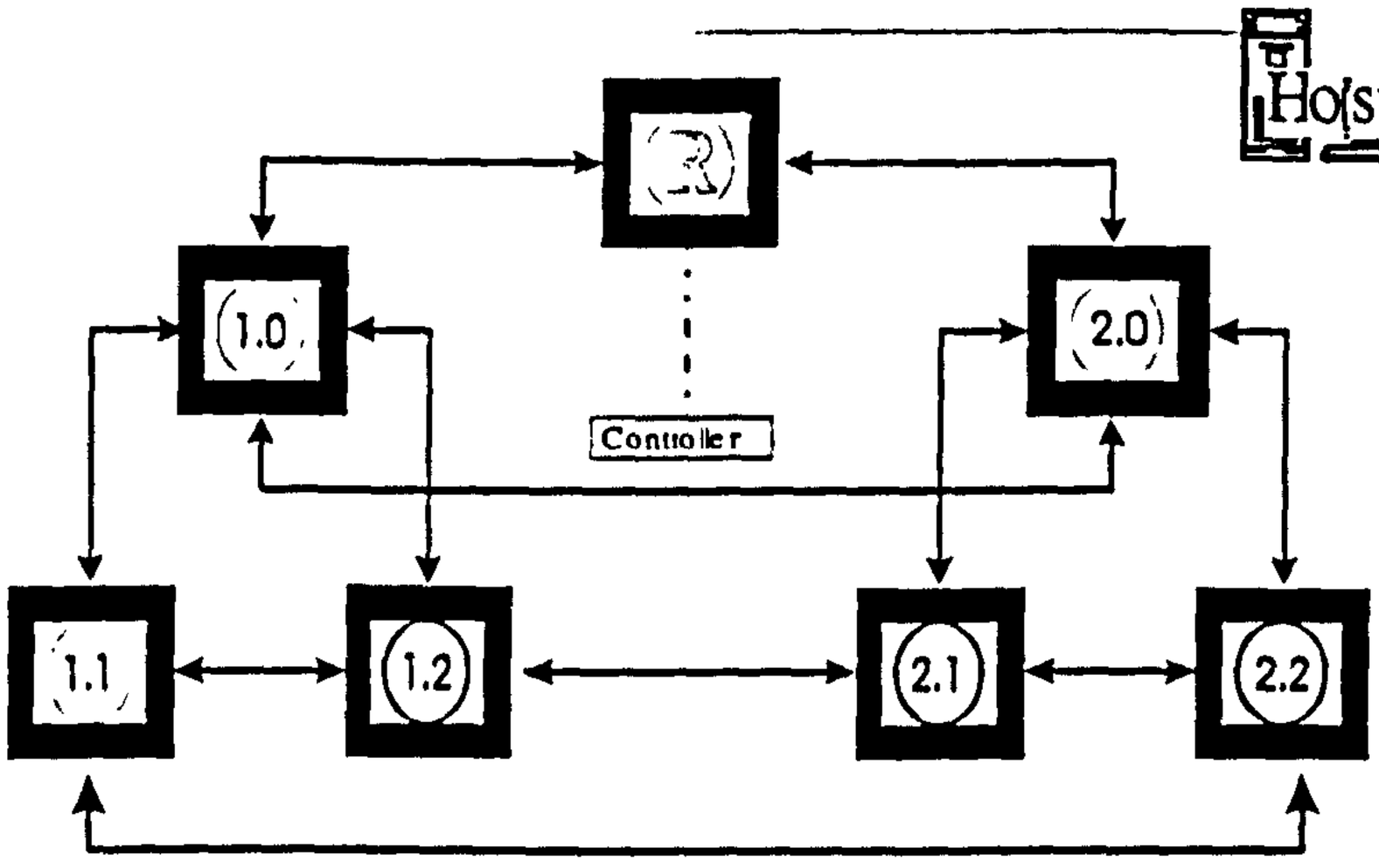

Figure 22: Transputer network

The structure used allows a pipeline implementation of the network using the internal ring (R, 1.0, 2.0). or the external ring (R, 1.0, 1.1, 1.2, 2.1, 2.2. 2.0). It also allows for each layer to be mapped on a set of transputers, dividing

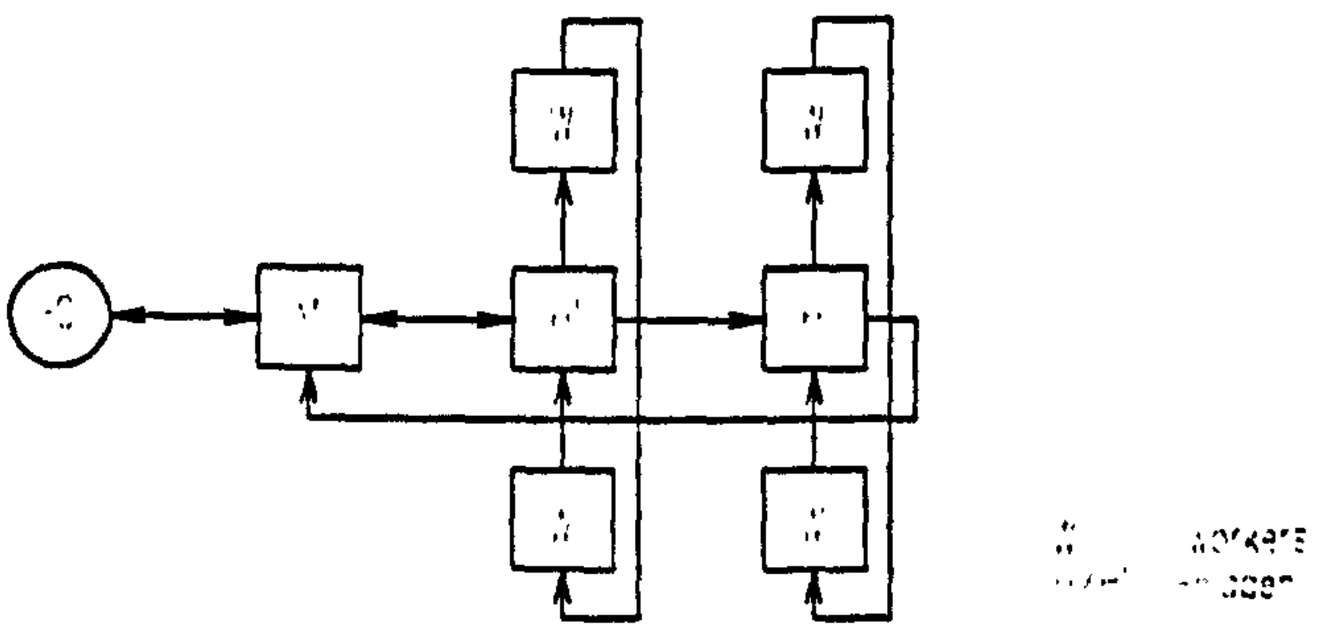

Figure 24: Process structure

between them the coupling matrices and multiplication procedures. resulting in a speed up in computation time. This has been done for both layers using 3 transputers for each of the H processes. From Figure 22 the pipeline runs

between R, 1.0 and 2.0. The layer processing H is split between another 2 transputers. 1.1 and 1.2 for the first layer, and 2.1 and 2.2 for the second layer. Transputers 1.0 and 2.0 run the same program as the program loaded on transputers 1.1, 1.2, 2.1 and 2.2.

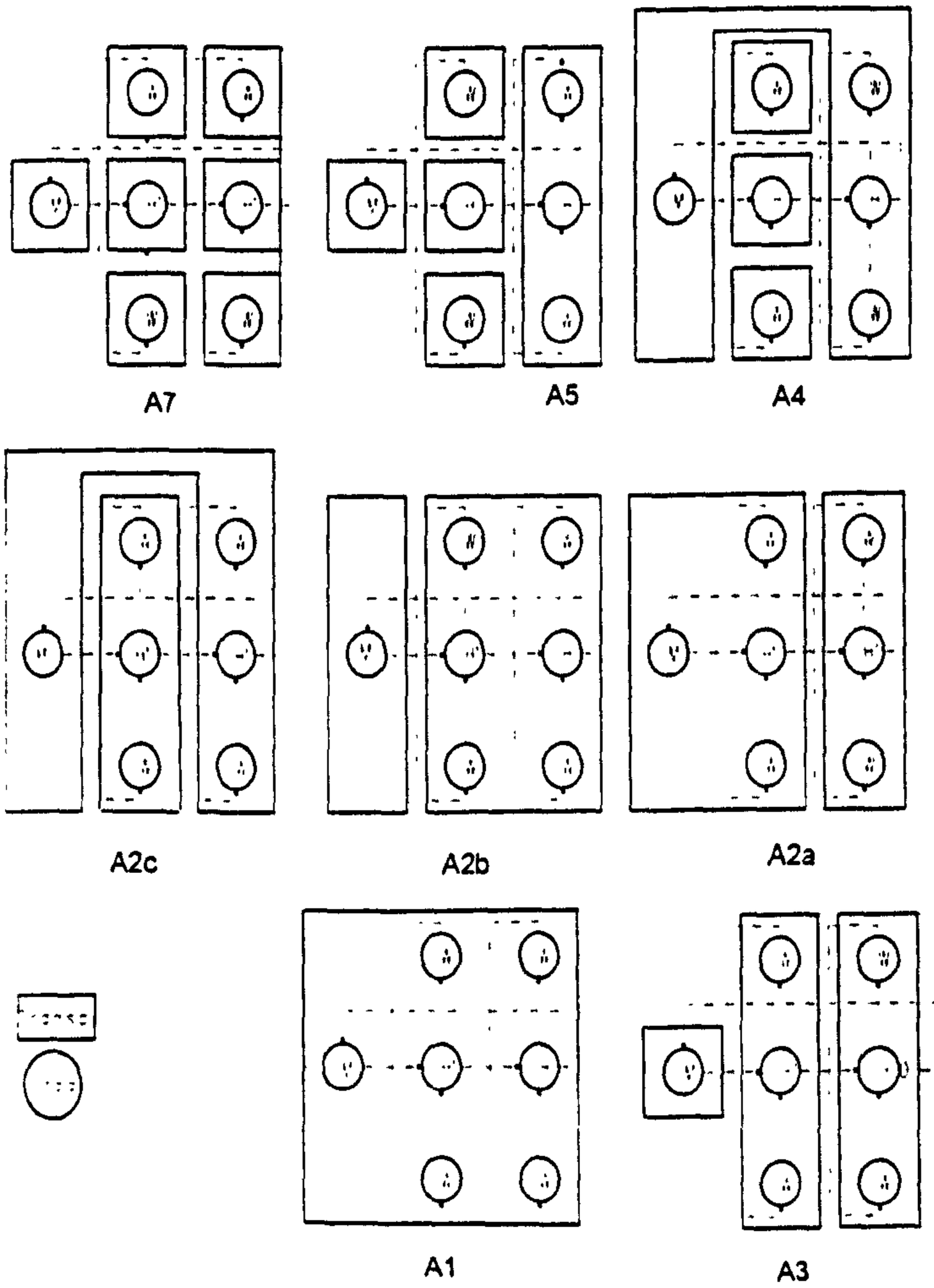

Figure 23: Allocation of processes on 7,5,4,2,2,2,1,and 3 transputers

Experiments were conducted again for the different possible mappings, through the use of the seven transputers and for several network sizes. as shown in Figure 23 . Here M is master process. and a partial Hidden layer process is designated by H'. This process is complemented by two parallel

processes designated as w. Process allocation is shown for each of the cases with the communication rings used. TABLE II shows the increment in time ratio, between the time used to compute x lines with n transputers and a single transputer running concurrent processes.

TABLE II
Efficiency allocation up to 7 transputers

		Process Allocation							
Size	Image	A1	A2a	A2b	A2c	A3	A4	A5	A7
18-20-1	A	1	0.61	0.56	0.83	0.5	0.39	0.39	0.39
18-20-1	B	1	0.69	0.54	0.77	0.54	0.46	0.46	0.46
50-25-1	B	1	0.78	0.75	0.96	0.75	0.49	0.48	0.48

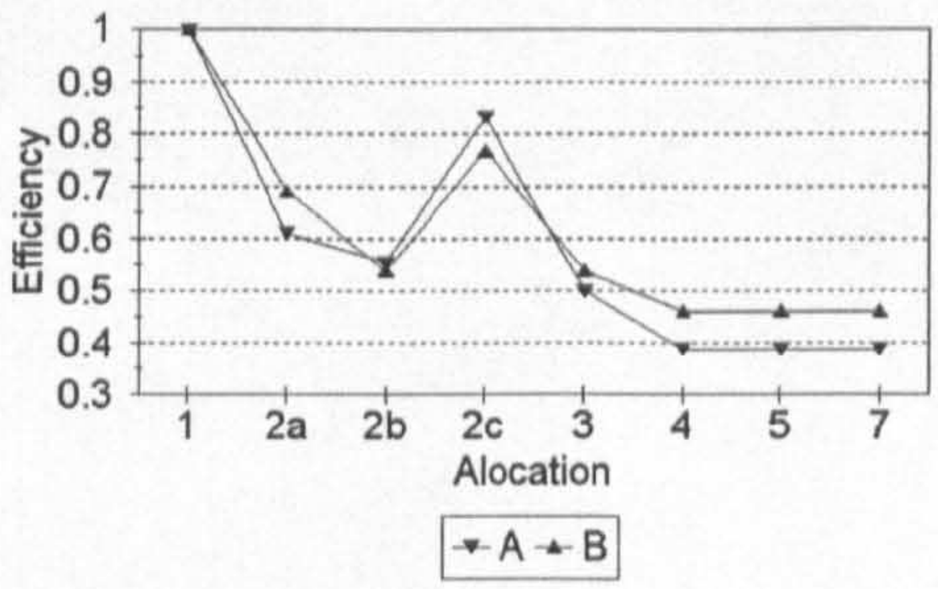

Figure 25:Time ratio allocation up to 7 transputers

From Figure 25 it is visible that the increment in computing speed is not proportional to the resources used. It is also clear that between 4 and 7 transputers no increase in performance resulted. This is due to the smaller size of the second layer used in all networks, since the output of all networks, is 1 node (omitted in the legend).

This shows clearly that a different allocation of resources is needed. In fact, the time used by the processes in pipeline extremities, should be smaller than the time for the middle layer. The alternative, is to reallocate resources from

this process, in favour of the middle layer. This can be done by allocating transputer 2.1 and 2.2 to the first layer, making a ring of 5 transputers to share the matrix multiplication.

This configuration is presented in the following section

4.4.3 Cross architecture

The structure described in this section is the reinforcement of the resources allocated to the first layer. We will refer to this structure as a *cross*, due to the shape of the schematic diagram. All four transputers that do not belong to the inner ring are allocated to the middle layer. The system works in this case based on a pipeline in the inner ring, with the four remaining transputers allocated to the first layer (Figure 26).

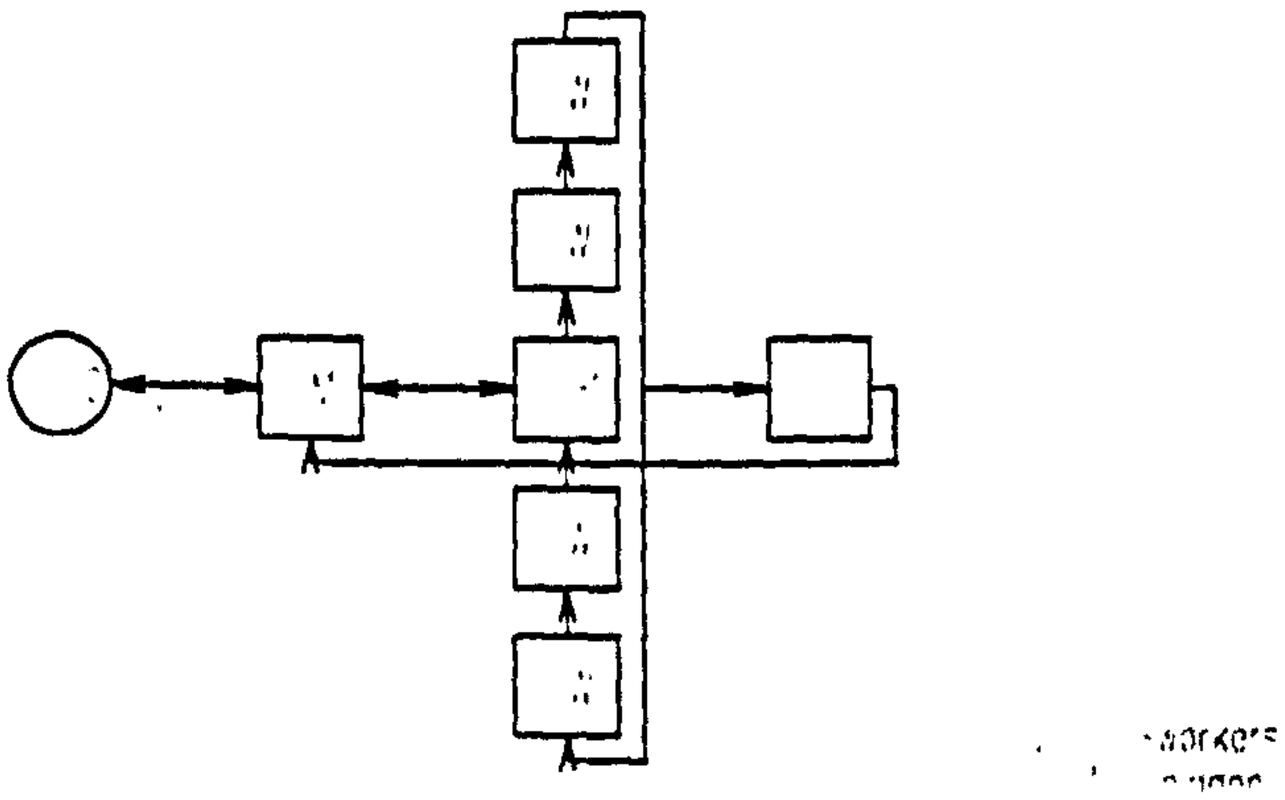

Figure 26: Process structure in *cross* architecture

The splitting of the main matrix multiplication could be done between the different workers by lines (li) or columns (co), depending on the relative

dimensions of the lines and columns. Two different node allocation schemes, where the matrices are split by lines between the transputers, are shown as li and li' in Figure 27. These correspond to different transputer loads on the inner ring. Time ratios obtained are shown in Figure 27 and are related to the full pipeline on one transputer.

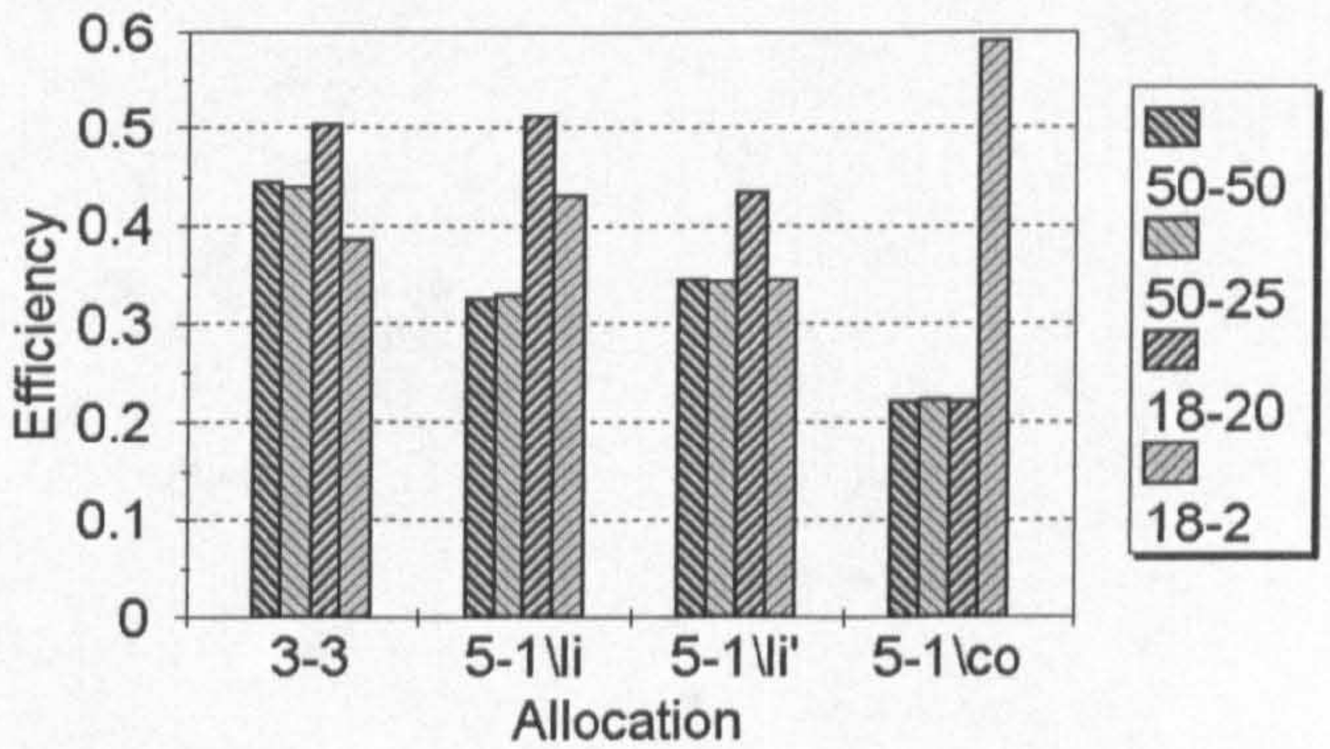

Figure 27:Time ratios for different allocation of lines between processes on 7 transputers

Although in the previous cases the best choice was independent of the size of the network, at this stage fine tuning is necessary to obtain the best performance.

To confirm the validity of the results, the program was altered to allow for Neural Nets having continuously increased sizes. Although these networks do not correspond to any solution, the time consumed to process several lines will be independent of the networks. Results are shown in Figure 28.

The first case is for a 50 input neural network, which corresponds to a 5^2 (^{10}) window working over two images. The values presented are the time in

10 This short form is used through the text as square windows are used.

seconds to output 9 lines of an image. It is clearly visible that with the exception of the very small networks, the column splitting option will be the best choice. This is due to the fact that it allows more efficient distribution of calculations, since the sigmoid function could also be distributed over the network used. It also highlights the fact that, in this case, the time consumed is not proportional to the size of the hidden layer. This is due simply to the allocation system used. Instead of an equal load, the system allocates *n/w* columns to each of the *w* workers, and the remaining to the main one. In this case 5 transputers are used which explains the period observed in the graph. Similar results are obtained for a smaller network.

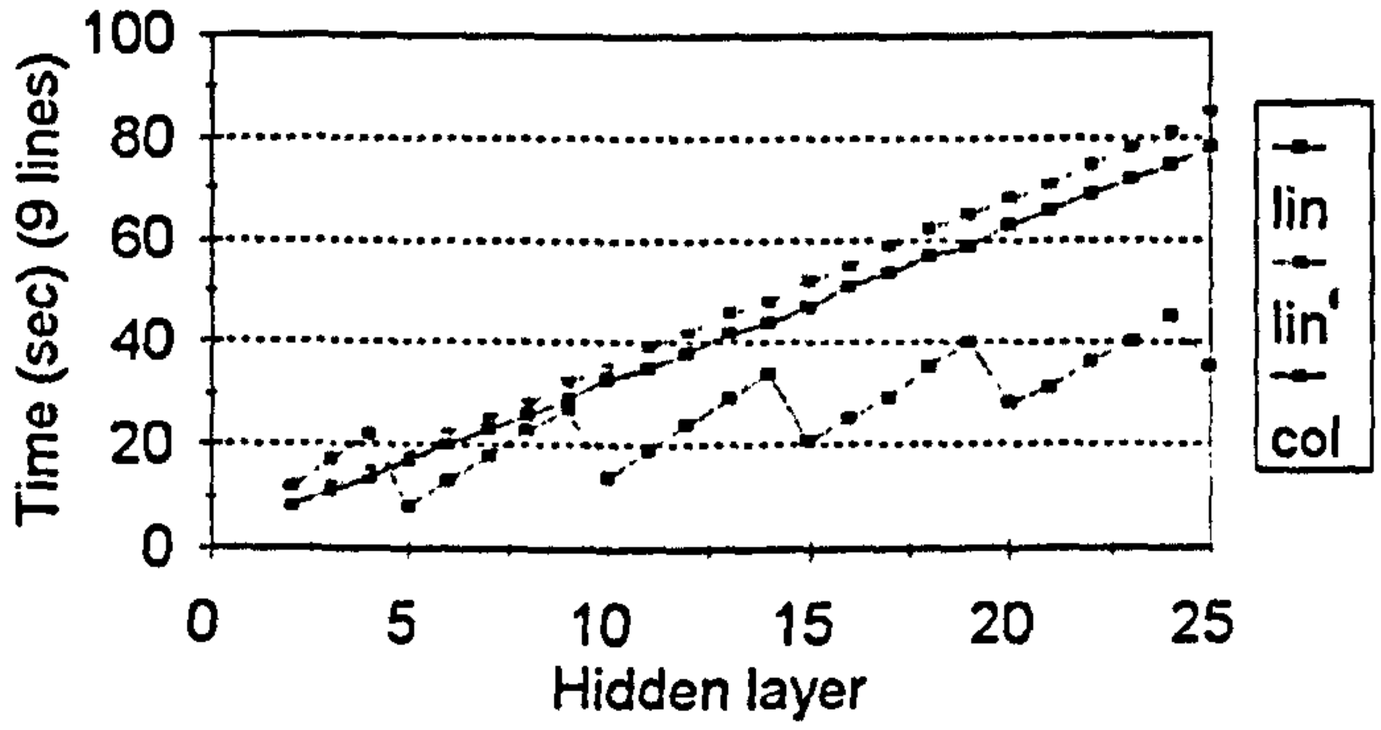

Figure 28: Processing time for different hidden layer sizes for the different allocations tested (Neural network with 18 input neurons and 1 output neuron)

4.4.3.1 Analysis

In this section the testing of several architectures, for the distribution of a neural network over several transputers, with emphasis on size is presented.

When splitting a simple algorithm over several parallel or concurrent processes an extra computational load due to the communications and spreading of the data will exist. This slows down the performance of the

concurrent version. The relation to the cheaper solution of 1 process on 1 transputer reveals that the economy obtained is substantial, the best results obtained being about 40%. The different results, related to the 1/1 solution, are presented in Figure 29.

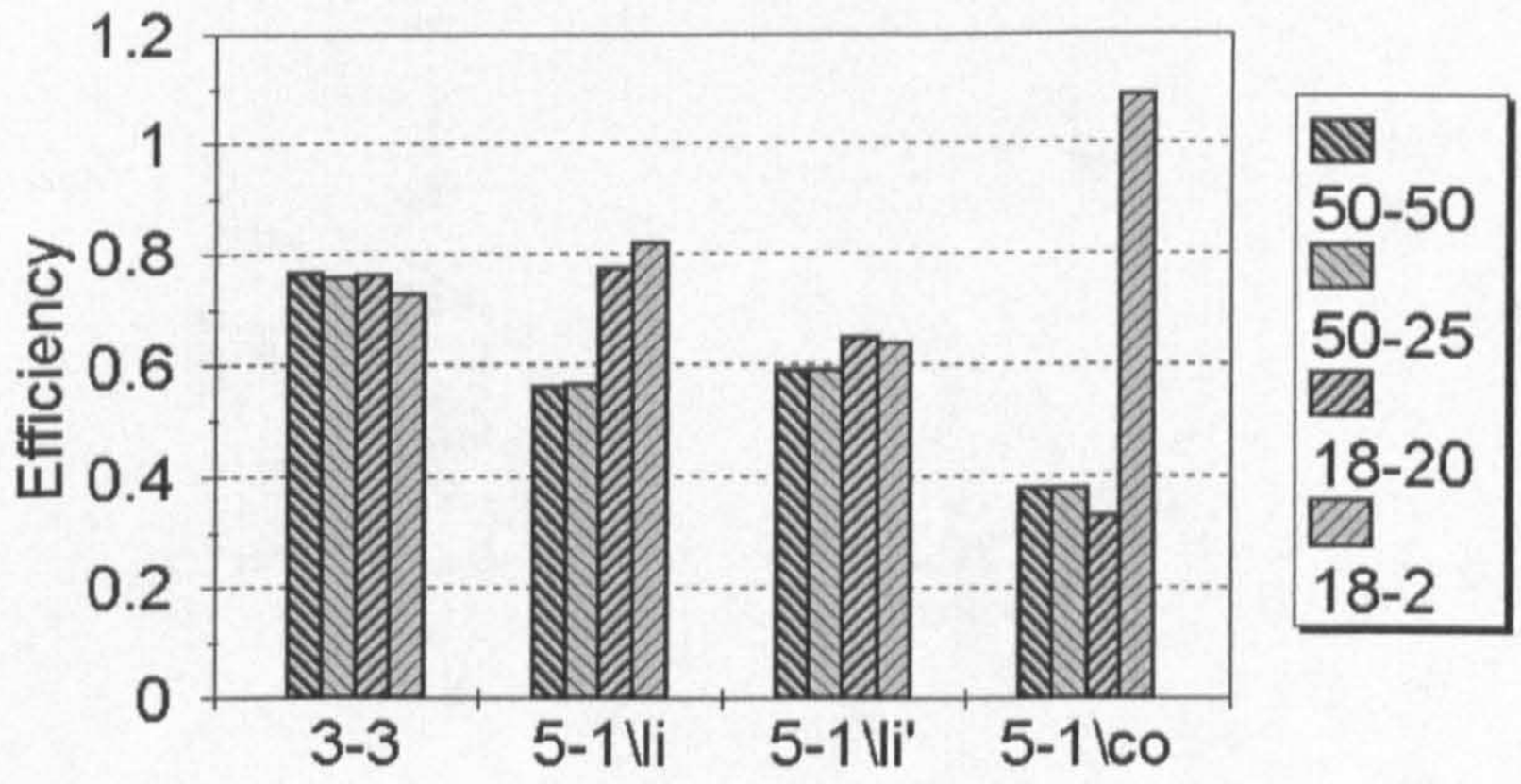

Figure 29:Time ratios related to one process

For the smallest network (18-2-1) , the last mapping of the multiplication by columns in a 5-1 transputer structure appears a little strange, in that the parallel solution is less efficient than the sequential implementation running on similar processors. This is due to the system that is used to assign columns to the different processors. Indeed, in this particular case we have the sequential solution, since all columns are allocated to the first transputer. Extra work is carried out due to the spreading of, in this particular case, meaningless vectors across the communication rings.

In this section we explored the capabilities of the pipeline implementation of the Multi Layer Perceptron. Although a general solution was pursued, the results obtained do not fulfil our expetactions. This is due to the amount of

values that are continuously flowing through the pipeline, which deteriorates the performance. Effectively, for each point in the image we are propagating intermediate real values, which is a time consuming activity. Although, the number of values could be reduced, a different strategy should be used.

4.5 Data Parallelism

In this section a data parallelism approach to the mapping problem is explored.

The parallelism is developed to the arbitration process, being the learning performed in a sequential system prior to the test of the solution. In this implementation each of the processors runs the full sequential program (which is more efficient than concurrent ones), with the same neural network, with each of the processors processing a subset of the image. This approach will allow a simplified strategy for the implementation of a complete system, since most of the edge detection algorithms are very suitable for a data parallelism implementation.

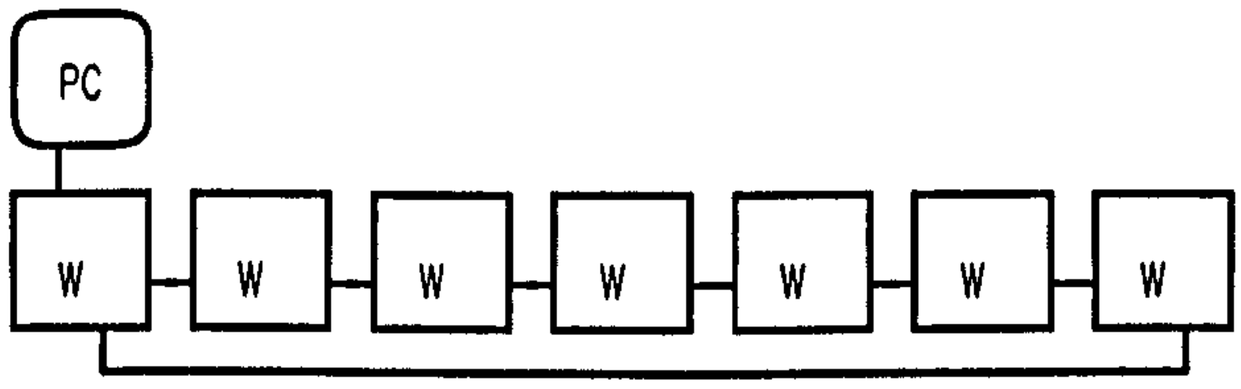

Figure 30 :Process structure for Data Parallelism

4.5.1 Structure Of A Data Parallelism Program

In the preceding section, the aim was to spread tasks over several processors,

in order to reduce the time used to perform the overall task. The aim now is to reduce the size of the overall task, spreading over several processors the data to be processed (Figure 30). This is accomplished by dividing the image into an appropriate number of sections, which are processed independently. These are chosen as consecutive stripes, allowing for the minimisation of communications time when dividing the image over the seven processors. The convolution nature of the algorithms used allows the processors to perform completely independent jobs, if they respect the condition that a margin is included in the image sections. The same algorithm is executed on each processor. In addition to the sequential program, procedures must be added to propagate the neural network coefficients and the images through the complete set of processors. The number of lines to be processed by each of the workers should be evenly distributed over the appropriate set of processors.

4.5.2 Implementation

The communications procedures spread equal values over all processors in a linear way. All communications are done in the first and last sections of the algorithms. The network structure is the same as used before. The communications are implemented as a ring, to minimise the number of programs used. The first transputer communicates with the C interface. It sends data to the second transputer and receives back data from the last. As before, such a structure simplifies the expansion of the network.

TABLE III shows the times obtained using this system on 7 processors, and the corresponding time ratios obtained are presented in TABLE IV. To obtain these values, the same images and networks were used as before, although all values are now with reference to a full size image (256 x 256 pixels).

TABLE III
Comparison of execution times (seconds)

	Network Size			
	18-2-1	18-20-1	50-25-1	50-50-1
Single Process*	246.4	2,038.4	4,300.8	8,422.4
Data Parallelism	47	436	820	1,347.58
Pipeline**	156.8	672	1,635.2	3,180.8

* Values extrapolated from 9 lines

** Idem

TABLE IV
Comparison for Relative time ratios on 7 processors
(related to 1 processor)

	Network Size			
	18-2-1	18-20-1	50-25-1	50-50-1
Data	0.19	0.21	0.19	0.16
Pipeline	0.64	0.33	0.38	0.38

The values obtained show that better performance is achieved using the data parallelism approach rather than the pipeline approach. Both systems use exactly the same hardware. The data parallelism technique is used for the simulations within the remainder of this thesis.

4.6 Conclusion

In this chapter several architectures have been tested as to their effectiveness for the implementation of the Neural Network arbitrator. Firstly the transputer and the basic parallel processing paradigms were described. Next the implementation, following the natural pipeline structure, of a multi layer perceptron neural network and the allocation resources were investigated. Finally, the implementation was restricted to the problem being researched,

and the suitability of a data parallelism implementation investigated. Measures were taken of the efficiency of the various implementations .

The next chapter, will describe the behaviour of several edge detection techniques. Their performance, on diverse images, will be presented, and their behaviour and limitations characterised. In the second part of the next chapter, the development of a neural network for edge detection is described. The procedure followed is presented along with the tests performed. The performance of the operator developed is compared with the other described methods .

5 PERFORMANCE COMPARISON BETWEEN VARIOUS EDGE DETECTORS

"Naturae enim non imperator, nisi parendo"[11]
Francis Bacon

5.1 Introduction

In the previous chapter the implementation of a neural network system upon a parallel processing platform was described. Main parallel processing paradigms were considered, with regard to the algorithm to be implemented and a best implementation decided upon.

In this chapter some examples of edge detection techniques are presented and their performance evaluated. The objectives are twofold. Firstly, to act as a comparison standard for both the neural edge detector and for the neural arbitrator. Secondly, to allow a selection criterion to be defined between different edge detection schemes that will be used for the arbitration schemes presented in the next chapter. Properties of each method are emphasised and comparisons drawn between them. The methods implemented are briefly described (appendix A contains a full description of the implementation). The criteria and method used to train neural networks for edge detection are

[11] Nature cannot be ordered about except by obeying her

presented, along with examples of their performance. The motivations for the options assumed along with examples are presented. Experiments performed to evaluate the solutions obtained are also presented.

5.2 Performance comparison of various edge detectors

5.2.1 Introduction

The first part of the work to be carried out was the implementation of edge enhancement operators, suitable for inclusion in the arbitration scheme to be developed later. These were implemented in Occam, as procedures supported by a C interface, and were similar to the ones used for the neural network implementation.

5.2.2 Images

All the tests presented were carried out on standardised images of fixed size and containing 256 grey levels. The image size was chosen in order to preserve a suitable amount of detail without making them impractical to use due to their size and memory requirements. Although a smaller size image would make processing easier and save on storage, minimising problems with disk space that regularly appeared, the visual contents would be lost due to their scarcity of detail. The images used provide a common base to the experiments performed, and permit direct comparisons between any of the results presented. It is expected that this will minimise the appearing or disappearing of features, in the processed images, due to the sampling, resizing or contrast corrections performed by the printing system used.

The images are stored and manipulated as raw format files. They are transferred to a TIFF[12] uncompressed format by the addition of a suitable header [139]. These images are then converted to a compressed format in order to save storage space. Apart from this, images are presented without further processing, unless explicitly mentioned. The majority of the pictures are presented negated to give more clarity and to save on toner. This inversion uses a linear look up table where white corresponds to the value 0 and black to the value 255.

To process the image border several artifices are used (rotation, reflection, filling, extrapolation, etc.) requiring the extension of the image size in order to accommodate the algorithm requirements. Except for the cases where the exterior border could be extrapolated from the image, the processing generates discontinuities in the border of the image. These could originate transient effects in the border vicinity, thus they are not processed.

[12] Tagged Image File Format

Figure 31 :Lenna Image

Figure 32: Girl Image

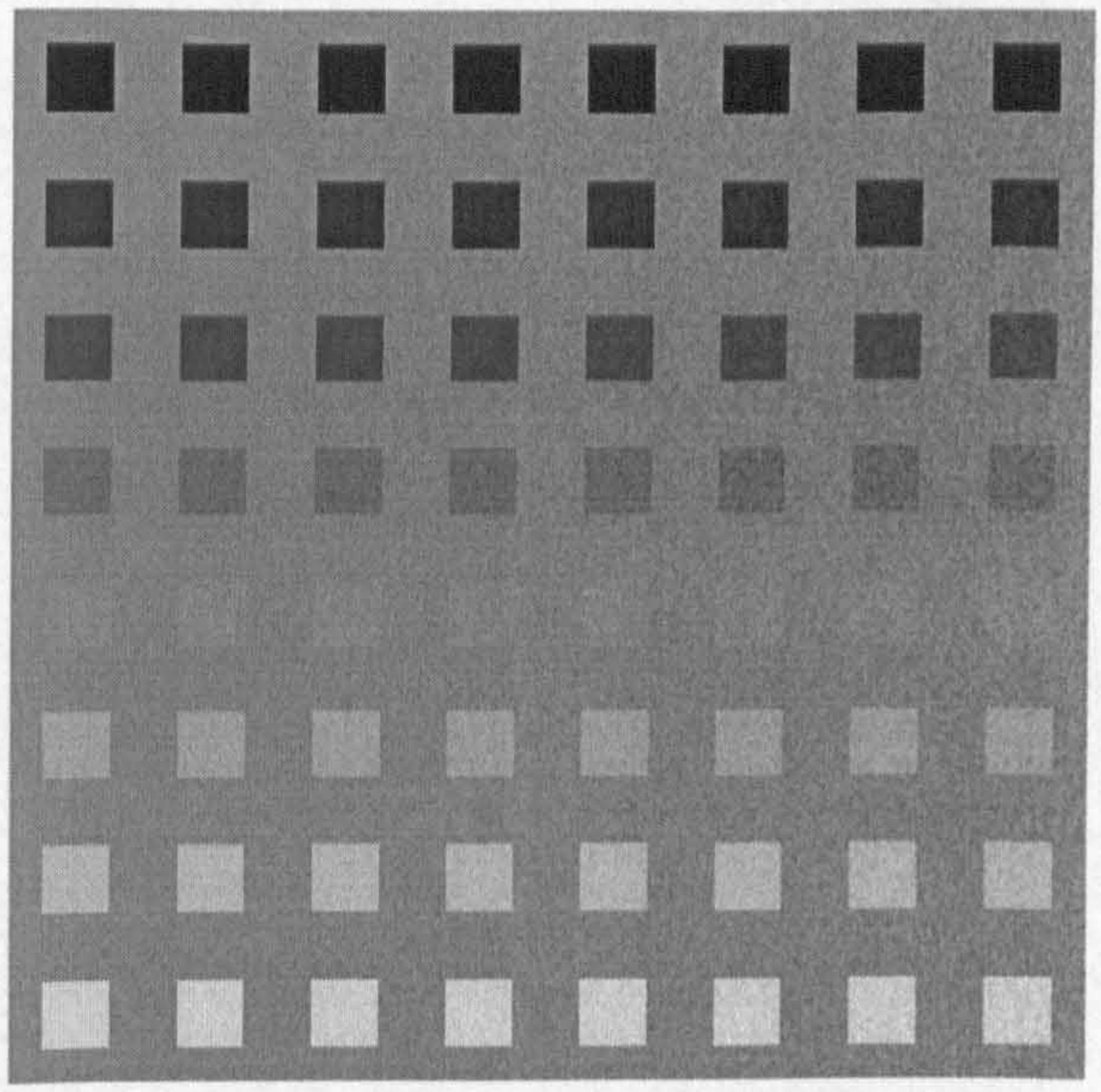

Figure 33: Squares image

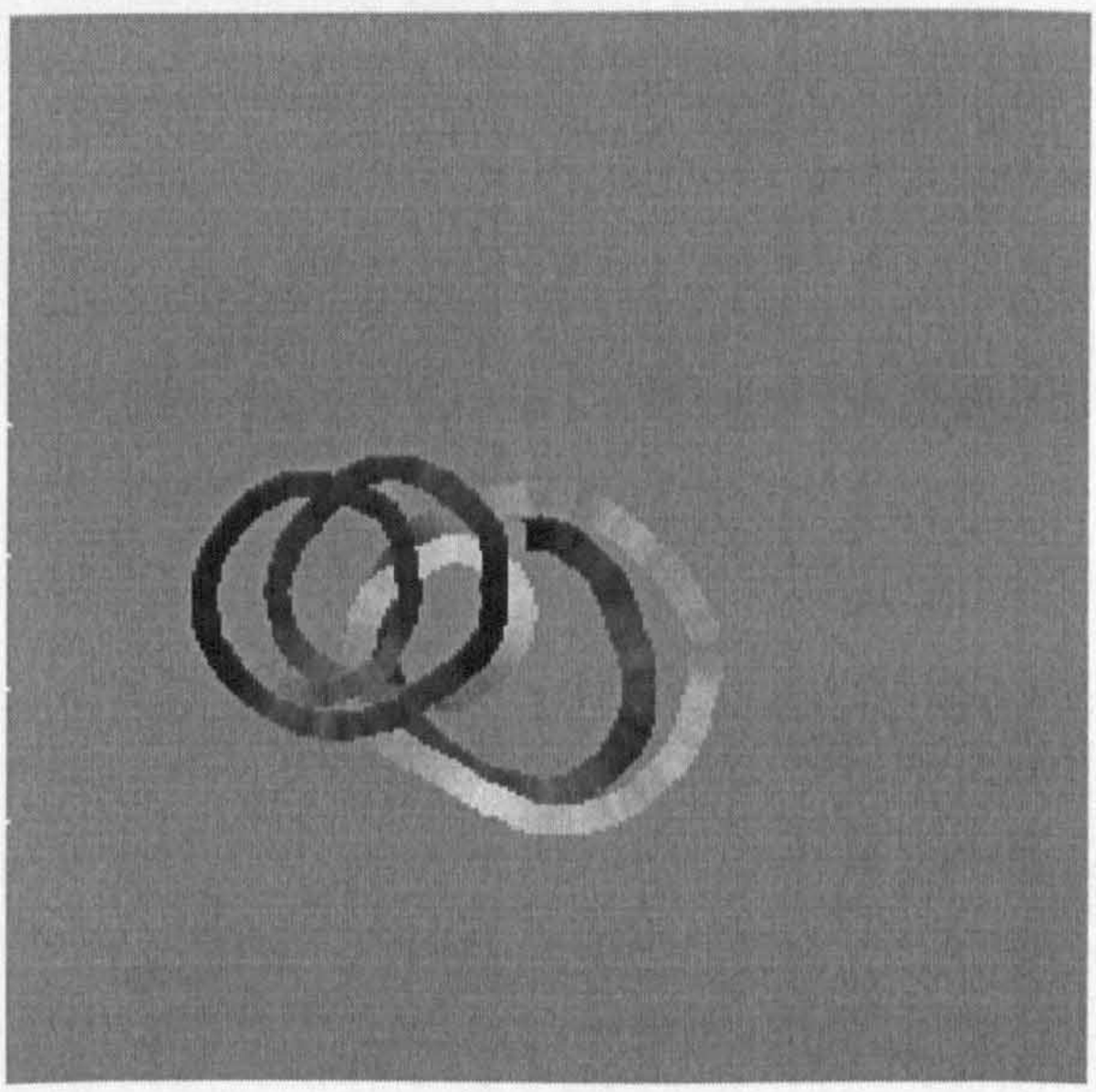

Figure 34:Band A

5.2.3 Image type

5.2.3.1 Image Selection

The selection of images capable of showing the performance of edge detection techniques is not a straightforward task. This selection is also highly dependent on the objectives or particular features that are required to be characterised. Several types of image are suitable candidates for the envisaged task, all presenting good and adverse characteristics. An image can easily be selected if some particular objective to the analysis is required. The selection of an 'average' typical image, if possible, is not an easy task. The basic images that can be used are artificial images. Examples of such images are the steps of different grey level amplitude and orientation as used by Pratt in the definition of his figure of merit (see section 2.1, [137]). Although these could be generalised for squares or circles of different amplitude in order to widen the edge orientations covered, they will lack some of the adverse features that are present in some real images. Their biggest advantage however is the fact that they can be used as patterns, since they are completely defined, as the edge position and shape are known. If the images are simple enough position errors can be easily computed, as error direction in marked edges can be unmistakably defined. This is, probably, the reason why line steps are used in Pratt's figure of merit. An example of such an image is shown in Figure 35, consisting of a single vertical step edge. If a bias in the edge position is introduced, by an edge detector, then this is measurable in the horizontal direction. To these pictures a small amount of noise could be added, or the image could be blurred. However, such images are unable to reflect some common problems present in real images, such as the behaviour of the edge detector when applied to more elaborate shapes.

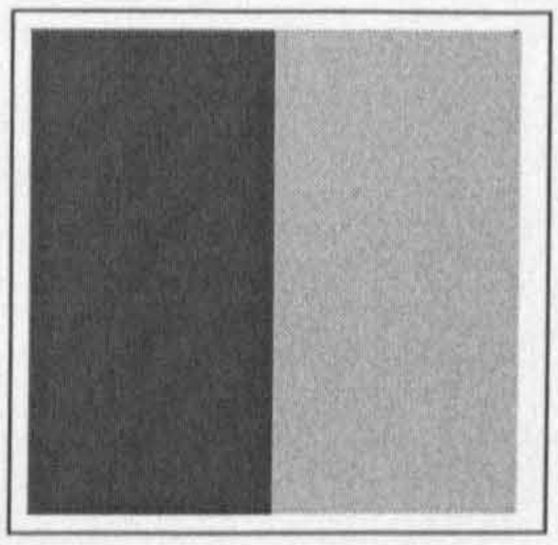

Figure 35: Step edge

This problem can be overcome by the use of more elaborated images. However, the simplicity when computing the edge bias would be lost. Images with more elaborate characteristics are used throughout the thesis (Figure 34, Figure 90, Figure 112). These image, which we will call Band A, Band B, and degraded Band A, are artificially generated. They consist of a stripe with a randomly varying radius and a randomly varying grey level slope. Figure 112 is a degraded version of Band A, blurred, and with added noise.

These images have a background grey level of 128. In all of them the stripe covers a wide range of grey levels with a reasonable distribution. Figure 36 presents the histogram of Band A, with the background points attenuated, in order to enhance its shape. Band A and Band B form a pair of images which have an important role in this work, as they act as learning and testing patterns for the neural networks to be used later.

Some images have acquired a standard status due to their wide general use. Examples of these images are presented in Figure 31 and Figure 32, referred to, respectively as Lenna, and Girl. They are complex and thus it is difficult to perform any quantitative comparisons. Also, due to the non existence of an edge map, other than that produced by an operator, an absolute grading is not achievable by the use of such images. However, the use of such images allows for a rough comparison with published results. These comparisons can not be accepted directly and without limitations, since different image sizes or

auxiliary processing, not always completely specified, will affect the results and performance.

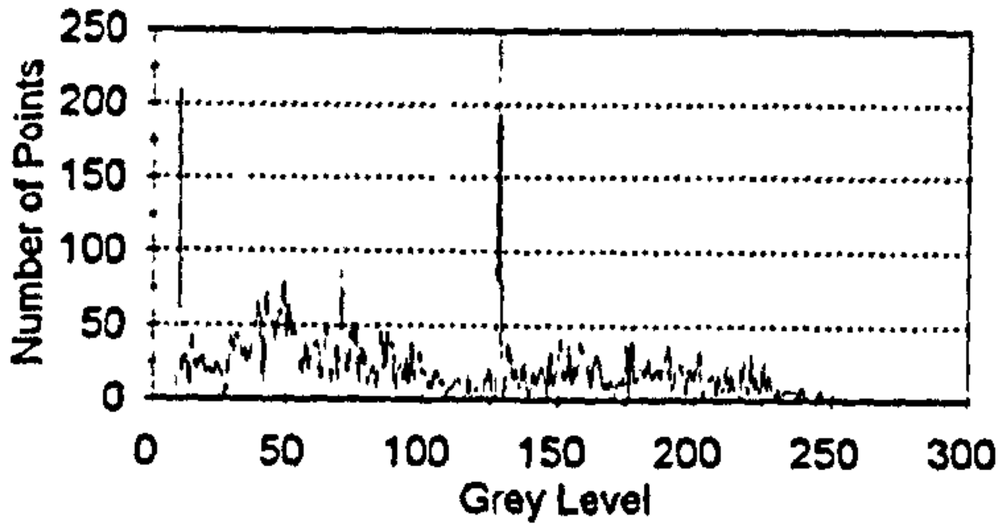

Figure 36:Histogram for Band A

A fourth image (Figure 33), also artificially generated, consist of squares over a background which has a grey level of 128. This image is intended to work as a gauge for different conditions, in terms of contrast and robustness to noise. The squares have eight grey shades ranging from black to white, from top to bottom, and the amount of noise varies from left to right by an amount proportional to the positional index. This image was preferred to the checkerboard image, used by several authors [61] [157] as it allows a wider range of contrasts between squares and background, and also allows the characterisation of edge positions.

However the final and most authoritative comparison should be drawn from the images to which the different methods will be applied. These images will contain all different imaging effects, favourable or unfavourable, to the same extent that the method or methods used are going to encounter in their particular application. It is intended that a general purpose tool is developed and thus a 'classical' set of images were chosen for the tests. To avoid errors that could arise from the interpretation of a particular image, more then one image is presented for each method.

5.2.3.2 Comparison criteria

The definition of a comparison criteria is dependent of the objectives. Our objective is the selection of sets of algorithms to be later used in the arbitration process. Different algorithms generate edge maps which are distinct enough to evaluate the differences between them. As a consequence of the different edge concepts cited in chapter 2, there is an inability to define the concept of important edges in a general context. Edges are perceived by observation thus the first comparison criteria is visual. Indeed, as we are able to understand the image and the 'objects' represented, it is an easy task to mark and evaluate, although subjectively, the completeness and quality of an edge map. This is the 'only performance measure of ultimate importance' according to W. K. Pratt ([137]). The fact that we are able to identify the objects presented in an edge map through our ability to evaluate the shape of the produced edges. makes this type of analyse more significant than a statistical technique. that does not take the shape into account. Although some quantitative criteria related to a particular feature is useful for an accurate grading, of edge maps or edge detection schemes. when adopting a general purpose approach they are insufficient. As a general purpose tool is intended, the first comparison criteria will be visual. However, it is important to easily distinguish differences in the edge maps produced by the different edge detection algorithms. This process involves the operation on a pixel to pixel basis. Common operations performed are (for two images I_A and I_B , and where w is a weighting factor. with $0<w<1$) :

Subtraction : $(I_A(x,y) - I_B(x,y)) + 128$ (50)

The shift is done to allow positive and negative values to be displayed with opposite tones. Pixels with equal values in both images will appear as middle grey (128) in the subtracted image .

Blending : $I_A(x,y) \times w + I_B(x,y) \times (1-w)$ (51)

The weighting factor *w* used is 1/3. This process is used for binary images which are defined with grey levels of 0 and 255. In this case common points reach a grey level of 255 and non common points, grey levels of (1/3)*255 and (2/3)x255, respectively for the first and second images. The resulting image could be enhanced by the use of false colours to distinguish small differences.

The value of 255 is obtained by thresholding, as binary images are usually stored with an intermediate grey level (128). The reason for choosing an intermediate value is related to the display of the images on the screen. If a high value was chosen then lines would be displayed with a very high level of intensity, which results in smearing and loss of definition.

A third method uses the addition of images:

$$\text{Addition: } a \times I_A(x,y) + b \times I_B(x,y) \tag{52}$$

where the constants a and b are chosen in order to increase the contrast of the resultant image. An operation (*'lighter'*)[13] that selects the brighter pixel :

$$\text{Lighter: } \max\left(I_A(x,y), I_B(x,y)\right) \tag{53}$$

However, some of the images obtained are very similar and difficult to grade or select just by visual analysis. Thus, as an aid, some measurements are carried out. These are done by counting the number of selected pixels according to an explicit criteria, which is particular to each of the images used. This criteria is based on the particular features, within the image, which reflect the performance of each of the edge algorithms being tested. Thus a particular feature is characterised and a figure of merit defined.

[13] For a direct Look Up Table, i.e., linear increasing values from black =0 to white =255

These values however do not take into account some important features of the edge maps (e.g. continuity, edge shape) and thus are used as a complement to the other methods. In the cases where the objective value is unambiguous the measures are normalised using that value. In the bands images a subtraction between the processed image and the reference edge image is done. This operation produces three types of values, negative, null and positive, corresponding respectively to categories where points are wrongly marked (W), correctly marked (C) and missed (M) in the edge map being analysed. The number of pixels in these categories are counted. As an uncertainty of the edge position could remain, a second reference image is sometimes used which is an enlargement of the thickness of the reference edge (3 points), referred to as an 'extended' image throughout the text. This map is produced by replacing each pixel in the edge reference map by a square of three by three pixels centred on the same position. As the values C, M and W have different ranges they are normalised by the length of the line used. using the ratio ()/r, where r is the number of points in the image used as the reference and () expresses the substitution parameter. This ratio is used so that values from the images can be compared on an equal basis. These parameters can however be misleading even in a normalised form, as a fully marked image will have all edge points correctly marked and a blank image will not show wrongly marked points. To avoid this problem other ratios are defined , designated as edge quality (EQ) and edge map quality (MQ) as follows

$$EQ = \frac{C}{W+C} \tag{54}$$

$$MQ = \frac{C}{W+C+M} \tag{55}$$

The map quality coefficient measures the number of marked points over the overall image related to the reference image. The missed points are included so operators that mark a small number of correct edge points are penalised. The edge quality coefficient refers only to the marked edges. TABLE V, A and B, represents reciprocal measurements, for Band A (Figure 34) and Band B (Figure 90) and can be used as a reference for optimal values in each of the cases. In Table V the number of points counted are also presented in a normalised form, where they are divided by the number of points that constitute the reference line.

TABLE V - A
Reference Values for Band A Edge Map Evaluation

Band A		Reference Values									
		Map					Ext				
		W	C	M	EQ	MQ	W	C	M	EQ	MQ
Map	P	0	1,323	0	1	1	0	1,323	2,811	1	0.32
	P/T	0	1	0			0	0.03	0.68		
Ext	P	2,811	1.323	0	0.32	0.32	0	4,134	0	1	1
	P/T	2.015	1	0			0	1	0		

TABLE V-B
Reference Values for Band B Edge Map Evaluation

Band B		Reference Values									
		Map					Ext				
		W	C	M	EQ	MQ	W	C	M	EQ	MQ
Map	P	0	1.622	0	1	1	0	1.622	3.337	1	0.33
	P/T	0	1	0			0	0.33	0.67		
Ext	P	3.337	1.622	0	0.33	0.33	0	4.959	0	1	1
	P/T	2.06	1	0			0	1	0		

In the case of the squares image it is possible to define precisely the position of the edges without a reference picture and as it covers a wider range of conditions it allows for the algorithm response to be measured in a more meaningful way.

The squares image is divided into the 64 squares. Each one corresponding to a different contrast and average noise quantity combination. For each of the squares the points in a neighbourhood from the correct position (±1 pixel in the direction normal to the edge) , the marked points inside the square and the points marked outside the square are counted. As the points outside the square have the same grey level the value obtained reflects the behaviour of the edge detector due to the noise only.

The non edge values ,inside and outside, are normalised by their maximum values, 240 and 720 respectively. The edge values are normalised by the precise length of the line measured in pixels (64). Optimal values are 0 for the outside area, 1 for the line and 0 for the inside area respectively. Values for the line coefficient greater than one reflects the width of the line. The maximum values attainable are different from the values above as neighbouring pixels are not counted twice. These are shown in TABLE VI.

TABLE VI
Reference values for the Squares image

	Line	Inside	Outside
Optimal	1	0	0
Real	3	0.77	0.9

For some of the methods investigated, details from the girl image are measured. In these cases a small area of the image, surrounding the detail, is

measured or the detail highlighted by changing the colour of the marked points These are then counted from the histogram of the image.

5.2.4 Methods Implemented

From the extensive set of algorithms presented in chapter 2 a selection of which to implement must be made. To facilitate this choice a number of books were utilised to catalogue widely used algorithms. These are summarised in TABLE VII. The practical problem of implementing and evaluating all the algorithms that have been proposed in the literature is beyond the scope of this work thus the choice of the most common used algorithm types was felt to be eminently sensible.

TABLE VII
Edge Detection Method Algorithm Selection

Ref Books \ Operators	Roberts	Sobel	Prewitt	Hueckel	Frei	Robinson	Prew temp	Kirsch	Laplacian	LoG	Marr	Canny	Others
Rosenfeld et al	×	×	×						×	×	×		
Whal	×	×		×									×
Vernon	×	×	×	×									×
Pratt	×	×	×	×			×	×	×	×			
Gonzalez et al		×			×								
Sonka et al	×		×				×	×	×	×	×		
Tested	⊠	⊠	⊠	□	⊠				⊠	⊠	⊠	⊠ □	

× Cited ⊠ Implemented □ Similar algorithm implemented

Classical local operators are preferred to ensure their practical usefulness. The selection is done with the purpose of covering different approaches. Widely cited methods were chosen to be implemented. It is assumed that the selection performed by each of the particular authors is based on criteria of performance, usefulness and representativeness of each particular approach.

They were also selected according to their implementation characteristics and similar approaches to the ones described are implemented, if they seemed more promising than the cited one. The Hueckel operator was one such case and it was replaced by the algorithm proposed by O'Gormann, which due to the use of square windows and Walsh functions seems to be more computationally efficient. A similar method to the one used by Canny, suggested by Deriche. was implemented as it allows for a recursive implementation. The complete and detailed description of the implementations are presented in Appendix A. The different methods implemented will be discussed in the next sections.

5.2.5 Derivative approaches

The first choice was to select the most natural approach. since edges are grey level changes the simplest way of detecting them is to measure the local gradient of the image. Of the algorithms presented in chapter 2 the Roberts, Sobel and Prewitt operators were chosen. as they are simple and computationally light. The localisation properties are a direct consequence of the definition and threshold used. They are able to mark the maximum of the derivative but require a search of the maximum in order to obtain a thin edge. as it is marked proportionally to the slope. The Roberts operator has an intrinsic half pixel error corresponding to the uncertainty in the centre pixel covered by the mask. The main disadvantage of this type of operator resides in the lack of robustness. due to the high pass frequency characteristic of the differential operator. Another practical limitation appears as a direct

consequence of this fact. As the edge selection is performed as a threshold process, the weakest edges are discarded together with the noise in the edge map.

The selected images processed by these methods are shown in Figure 37 through Figure 48.

Figure 37: Lenna processed by the Roberts Operator

Figure 38: Girl processed by the Roberts Operator

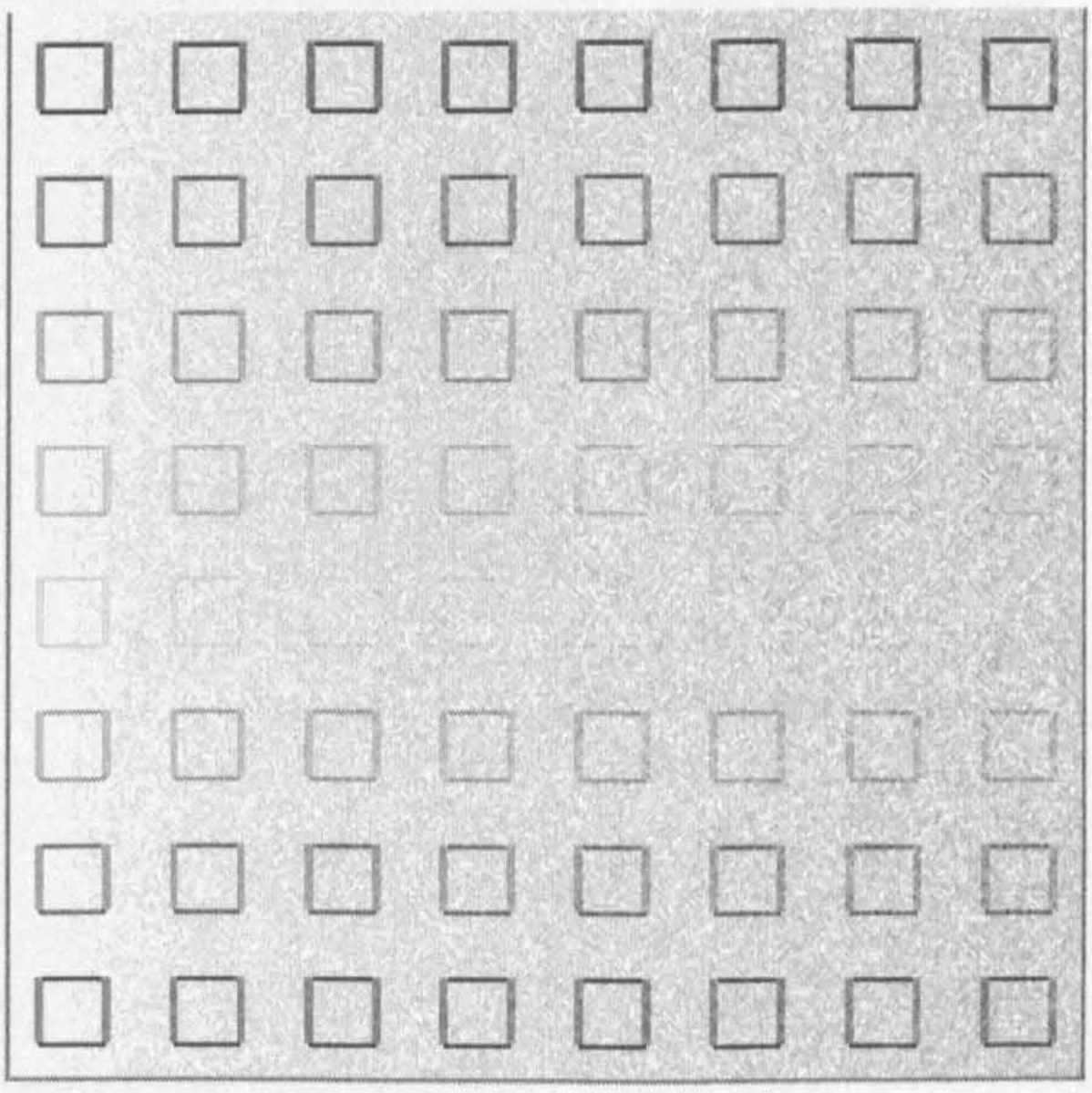

Figure 39 :Squares processed by the Roberts Operator

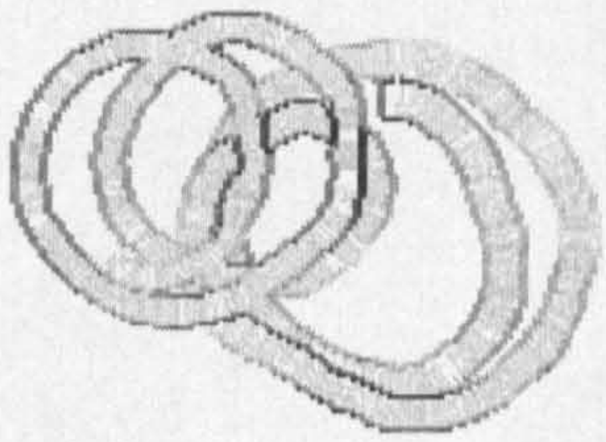

Figure 40:Band A processed by the Roberts Operator

Figure 41: Lenna processed by the Sobel Operator

Figure 42: Girl processed by the Sobel Operator

Figure 43:Squares processed by the Sobel operator

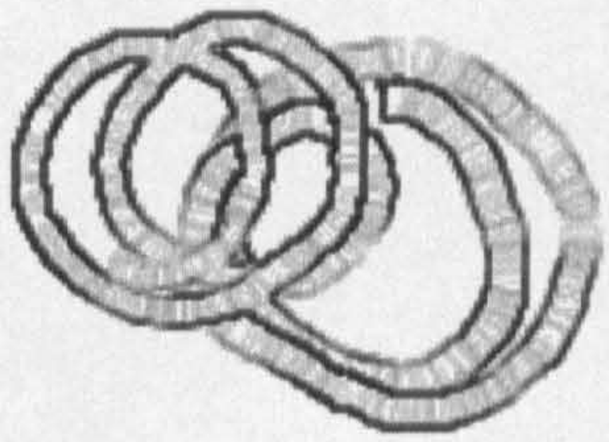

Figure 44:Band A image processed by the Sobel operator

Figure 45:Lenna image processed by the Prewitt operator

Figure 46:Girl image processed by the Prewitt operator

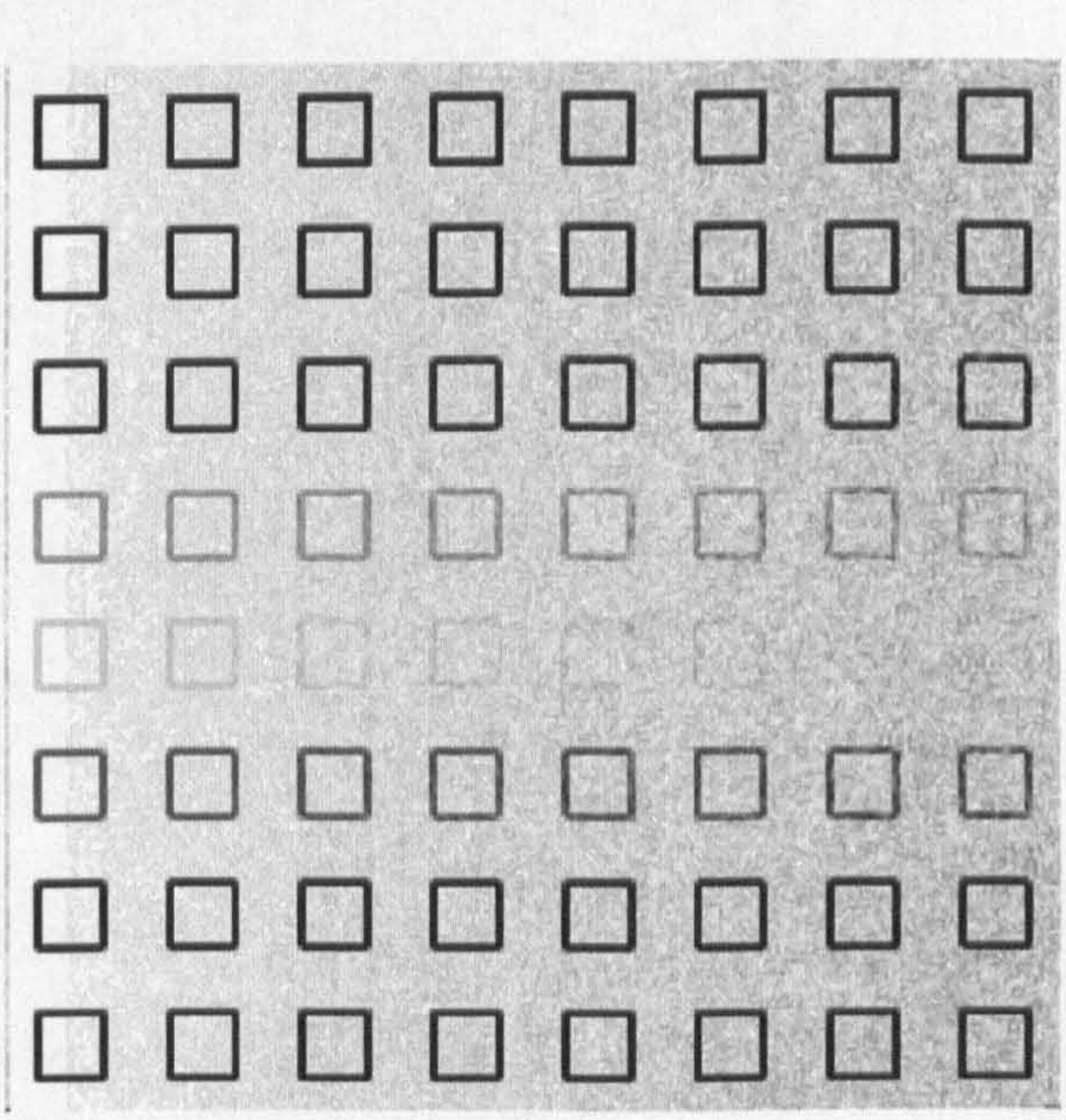

Figure 47: Squares processed by the Prewitt Operator

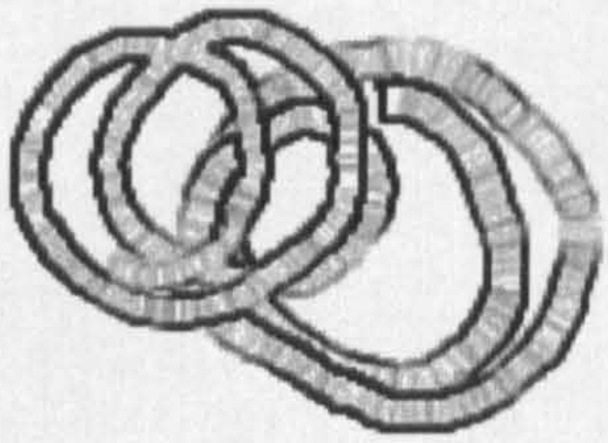

Figure 48: Band A processed by the Prewitt Operator

5.2.6 Templates

An operator which uses a formal distinct approach and which has a similar form to the derivative approaches implemented is the one proposed by Frei and Chen. Although formally distinct, it uses a set of convolution masks and a more elaborate threshold. Both complete and short forms of the technique were implemented, without any relevant difference between the results being noticed. The short version is used as it is more computationally efficient. The output of the short implementation of the Frei and Chen operator when applied to the selected image is shown in figure 49 through Figure 52.

Figure 49:Lenna processed by the Frei and Chen Operator

Figure 50: Girl processed by the Frei and Chen Operator

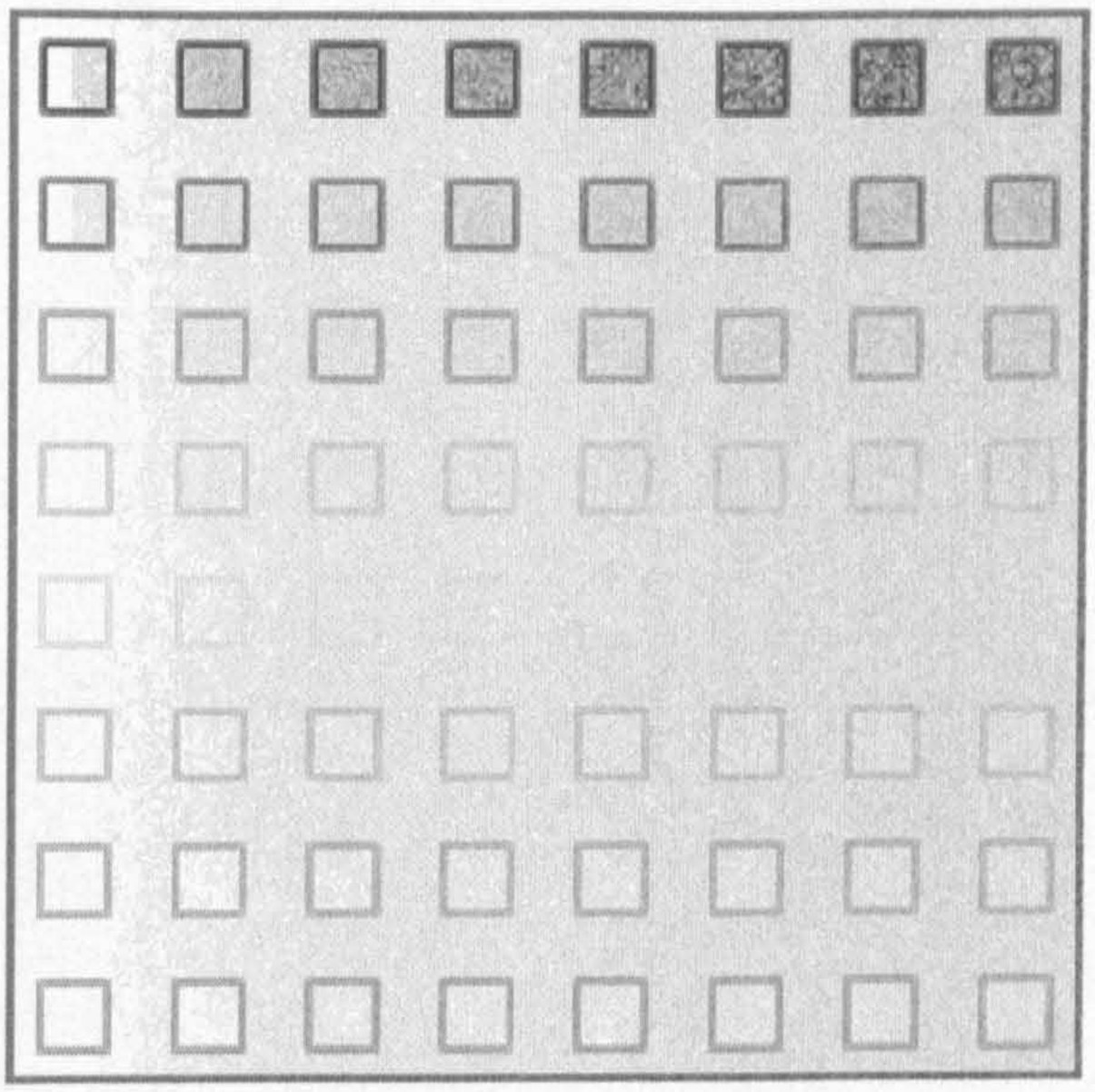

Figure 51:Squares image processed by the Frei and Chen Operator

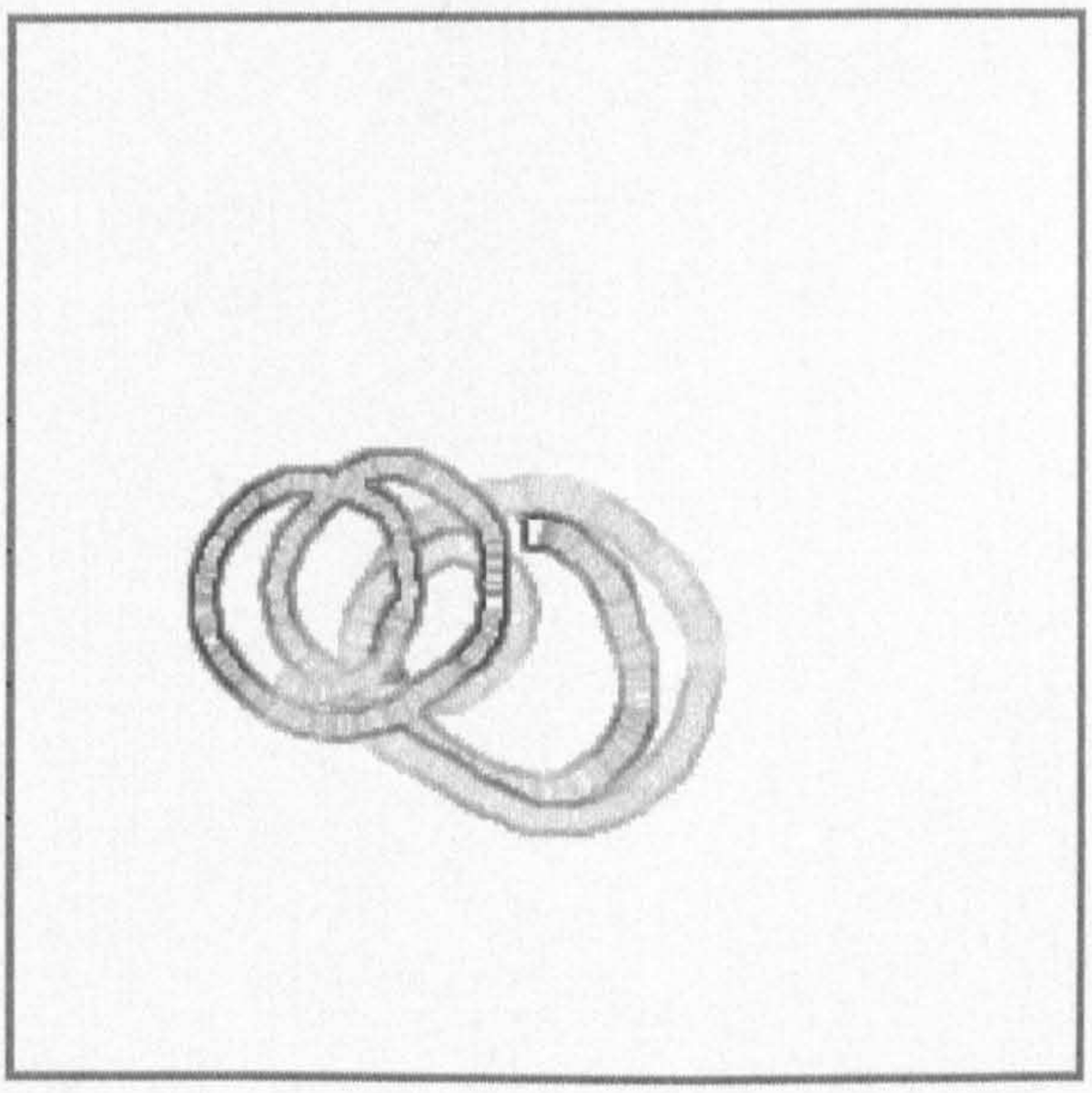

Figure 52:Band A processed by the Frei and Chen Operator

5.2.7 Second Order derivatives

The second group of methods investigated are the zero-crossing and optimal approaches methods. They have several advantages over the previous group, namely that they mark edges more completely and thinner. The main advantage resides in the production of one line thick edges. The edges produced form closed contours in most instances. Their main disadvantage is the fact that they produce a large number of residuals, due to their frequency response. An example of such operators is the Laplacian. It marks one pixel wide closed contours if no noise is present in the image, otherwise the amount of features originated by the noise makes the image useless.

This fact makes the Laplacian an inappropriate operator unless some form of smoothing is performed beforehand. In the case of the Laplacian of the Gaussian, or the related Marr and Hildreth operators, smoothing is incorporated in the filter itself. Although the problem of marked spurious features is partially solved, a bias in the edge position appears due to, and proportional to, the amount of smoothing performed. Two related variants were preferred and thus were also implemented. These correspond to optimal filters, and are the Canny and Deriche operators. They are quicker to execute, whilst maintaining some of the same characteristics, in the produced edge maps, as the LoG operators.

Figure 53 through Figure 62 shows the selected images processed by these algorithms.

Figure 53:Lenna image processed by the
Zero Crossings of the Laplacian of the Gaussian

Figure 54 : Girl image processed by the
Zero Crossings of the Laplacian

Figure 55:Lenna image processed by the Canny operator

Figure 56:Girl image processed by the Canny operator

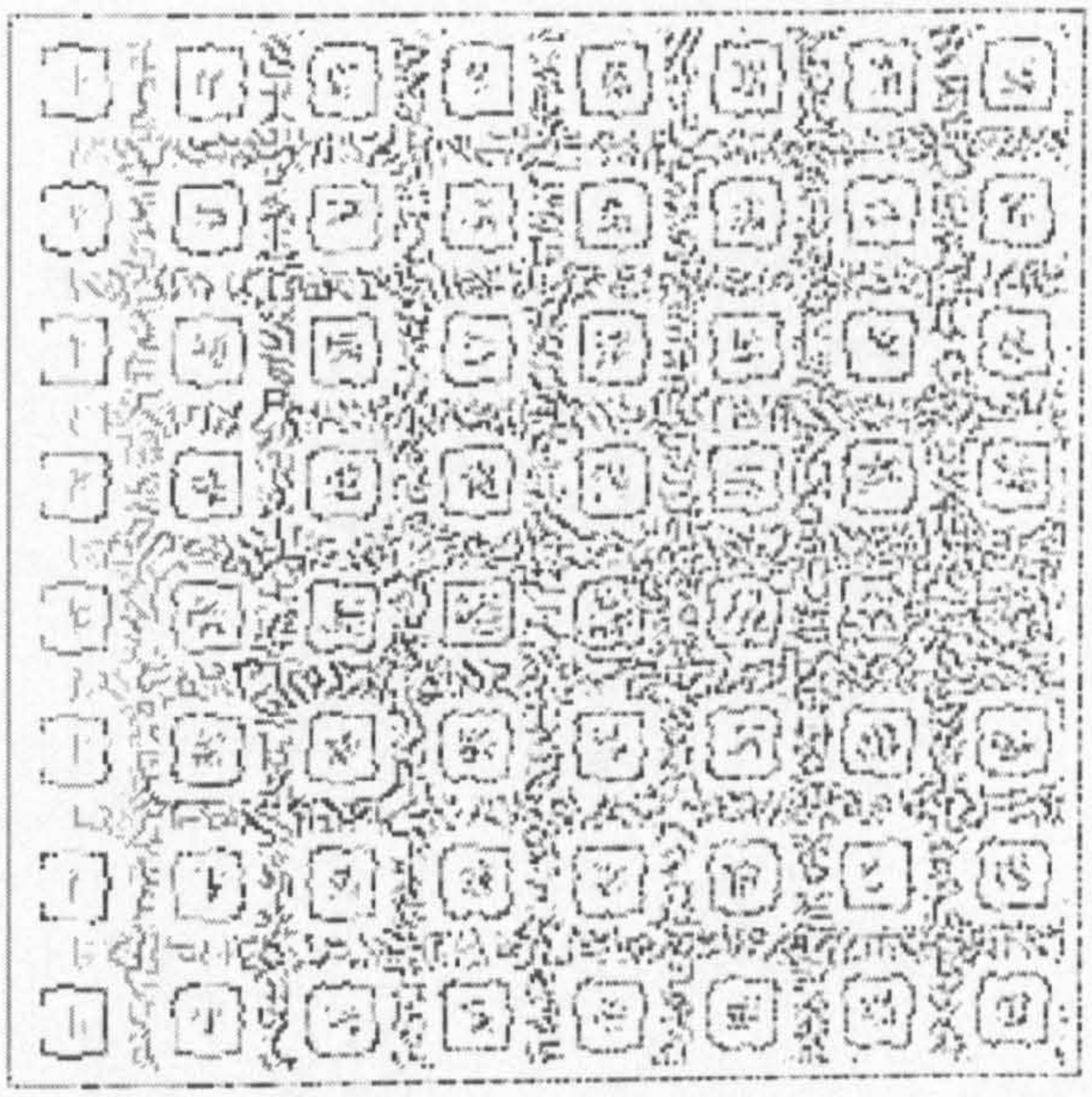

Figure 57:Squares image processed by the Canny operator

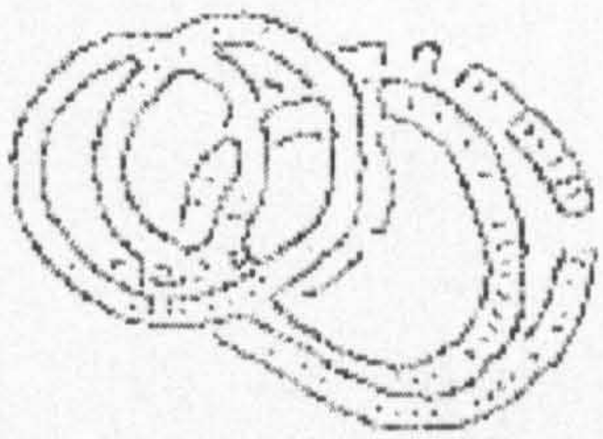

Figure 58:Band A image processed by the Canny operator

Figure 59:Lenna image processed by the Deriche operator

Figure 60:Girl image processed by the Deriche operator

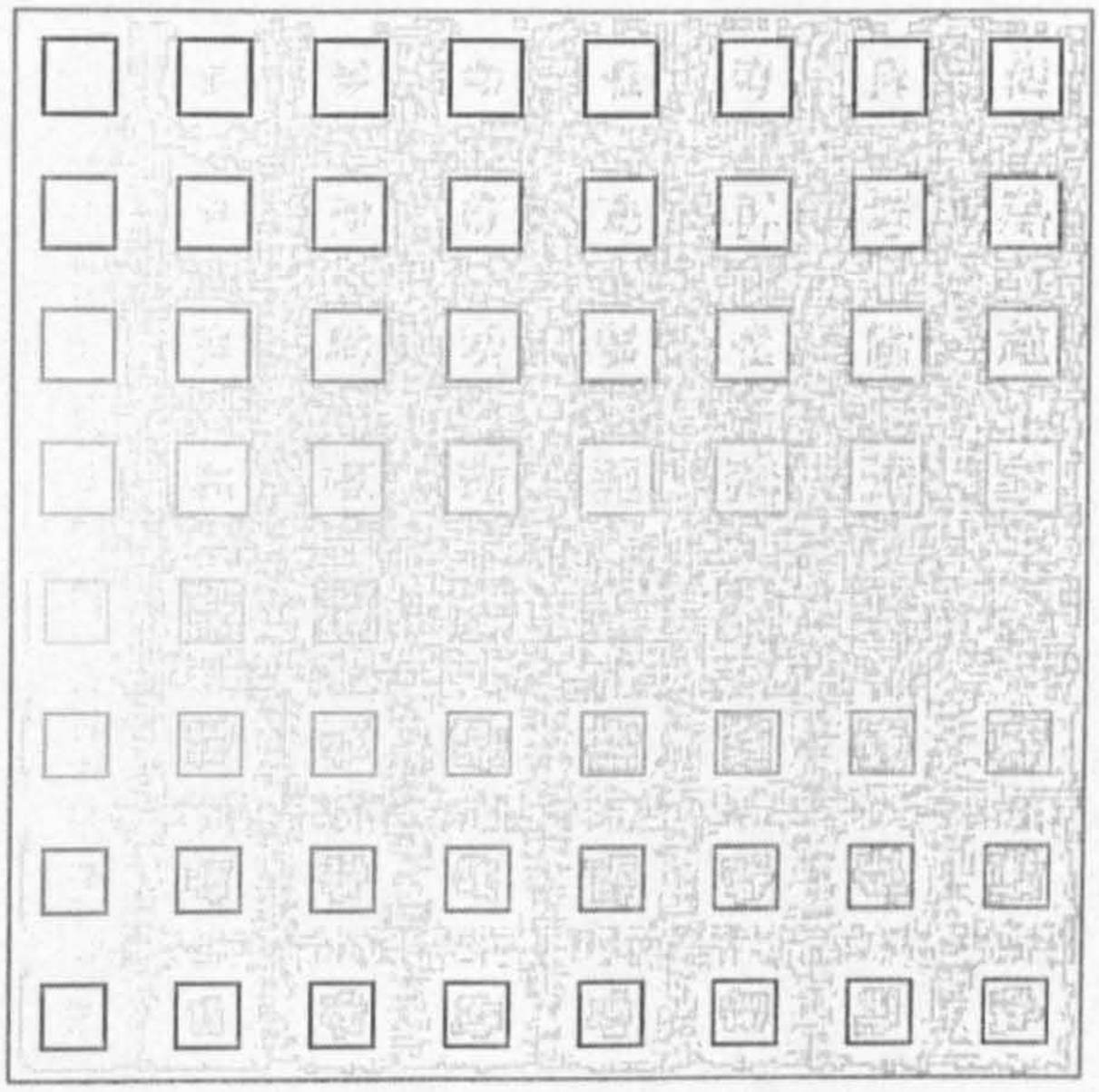

Figure 61:Squares image processed by the Deriche operator

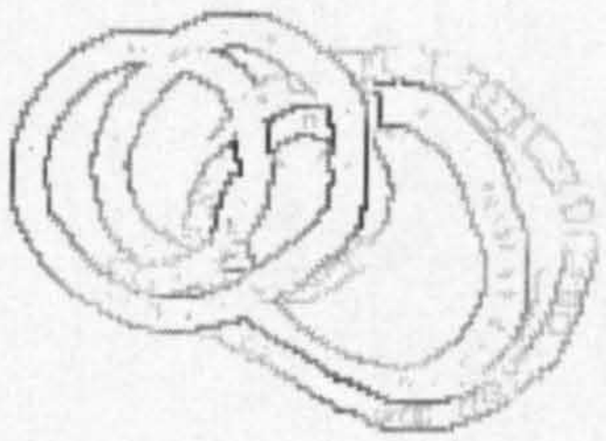

Figure 62:Band A image processed by the Deriche operator

5.2.8 Surface Fit

A third group of algorithms, known as surface fit, are also commonly cited. These fit a step to a window in the image. The first example was the Hueckel edge detector operator. A later version of such an operator was proposed by O´Gormann. The later was the operator that was implemented. The results of the application of the operator to the selected images are presented in Figure 63 through Figure 66. It was found to be the only operator which consistently marked all the edges in an image. Unfortunately, it also marks grey level ramps as edges, and the amount of points marked largely exceeds the number of expected edges in a common image. However, if the grey level gradients are small, it was found to be the method that presents the best amplitude invariance resulting in edges being marked from very low contrast images.

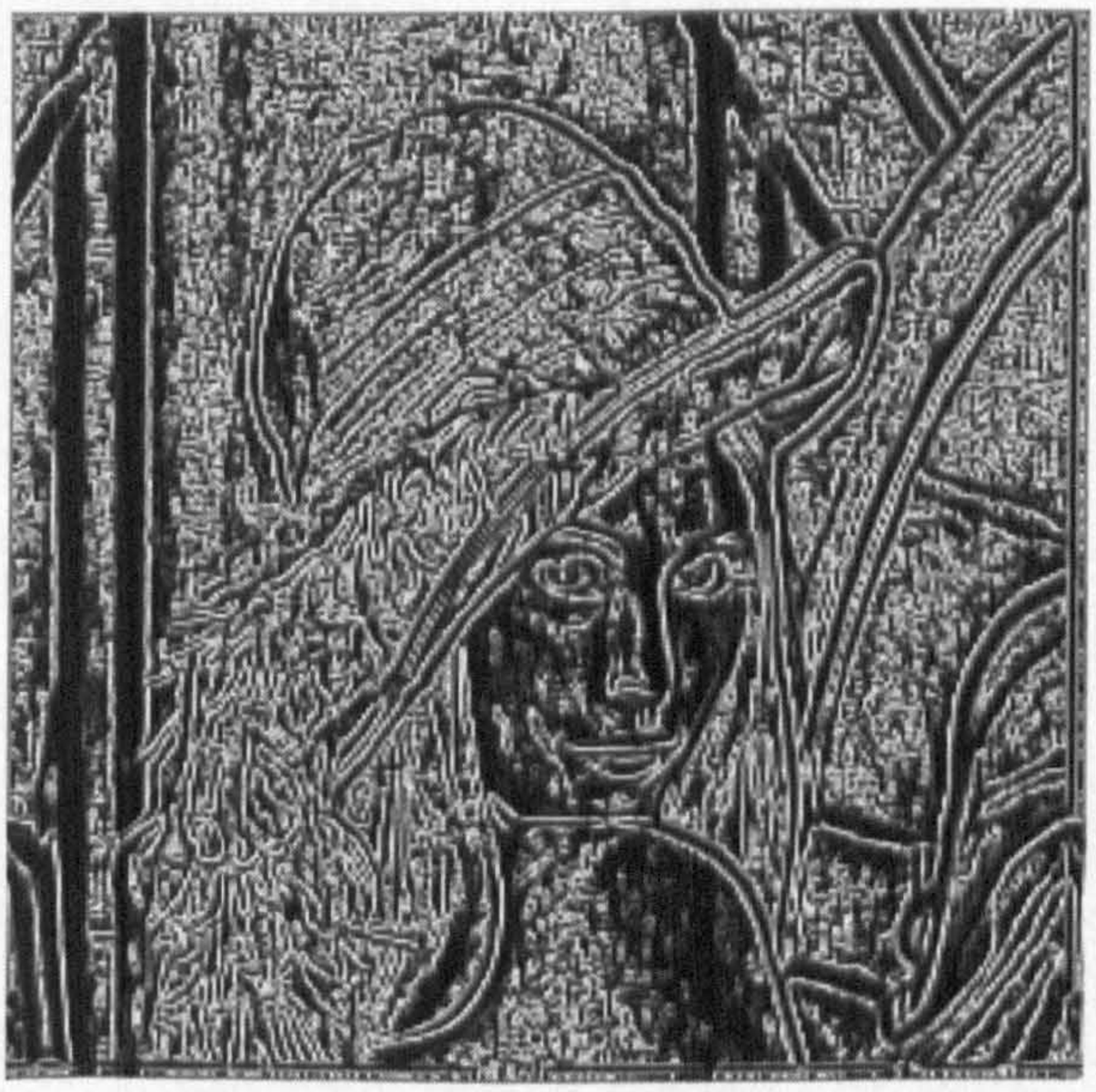

Figure 63:Lenna image processed by the O'Gormann Operator

Figure 64:Girl image processed by the O'Gormann Operator

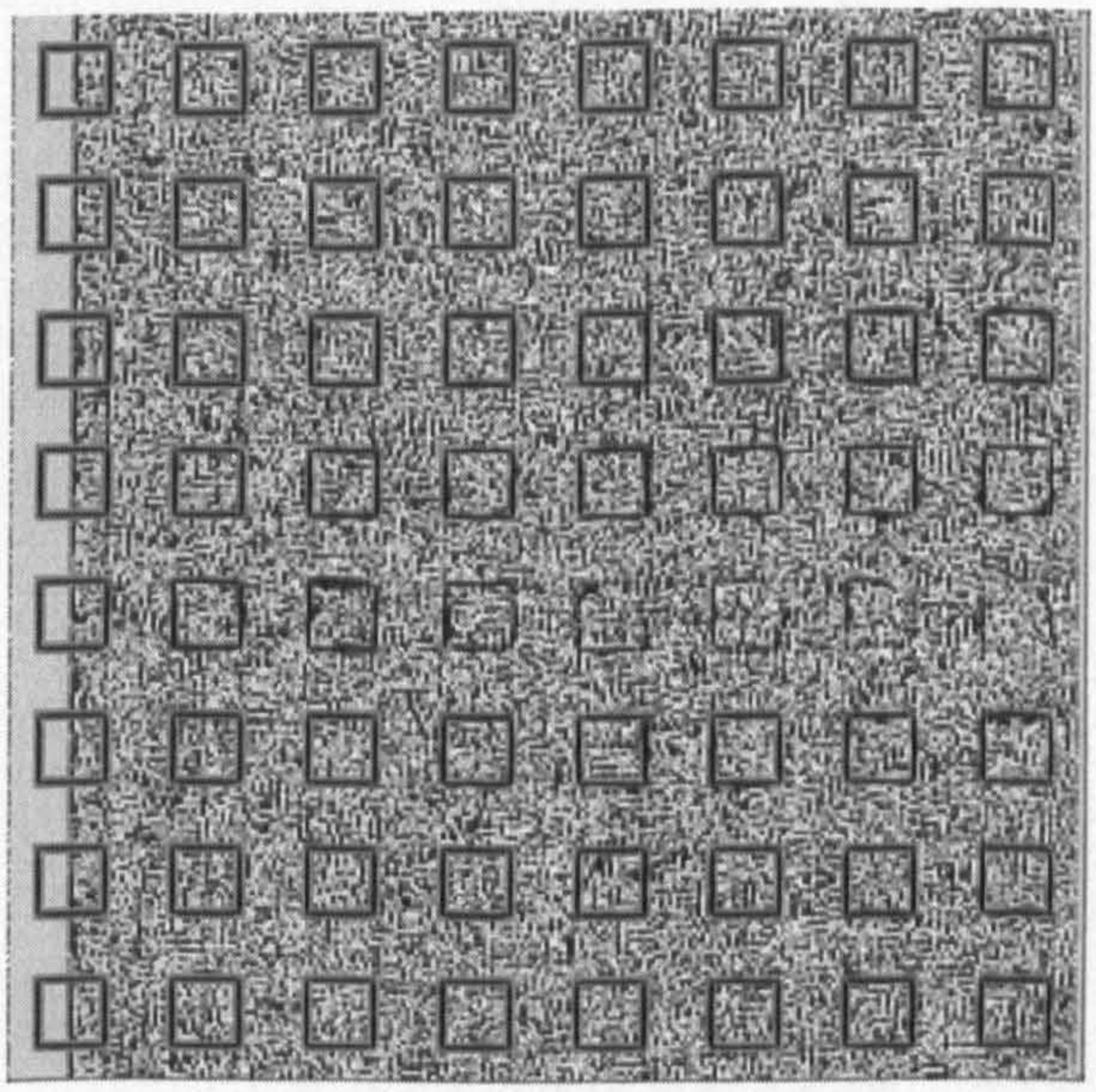

Figure 65:Squares image processed by the O'Gormann Operator

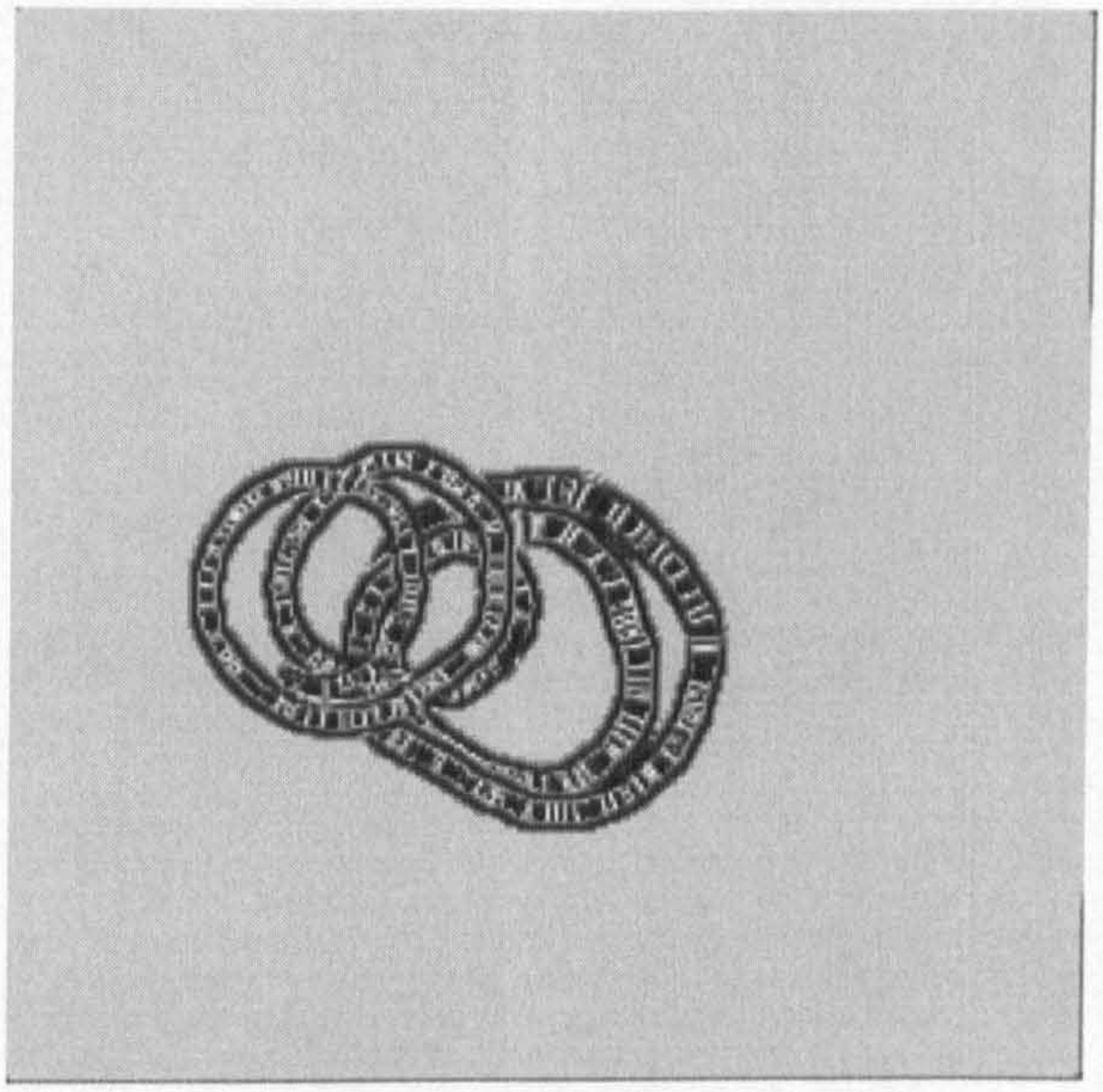

Figure 66:Band A image processed by the O'Gormann operator

5.3 Comparison

For all the techniques investigated objects are recognisable in the majority of cases The major differences between the techniques resides in their noise handling capabilities. A comparison between figures 37 to 66 shows a remarkable similarity between the different techniques. Indeed it is difficult to discover a line which was missed by any of the techniques except for the Roberts operator where the apparent poor performance is mainly due to the low contrast of the produced image. A simple change in the Grey level look up table will reveal the remaining lines (see Figure 67).

Figure 67:Lenna image processed by the Roberts operator with enhanced contrast.

However edges are marked with different intensity by the different algorithms. Examples of the phenomenon can be seen in the white column on the left in the Lenna image (detail A, see Figure 68), in the hat band (detail B), the window in the Girl image (detail C, see Figure 69) or the scarf (detail D). Particular attention is given to detail E. Although it is not clear if it is a background feature, or if it has another source (such as a reflection), it has a clear defined shape and it is situated in a non homogeneous background.

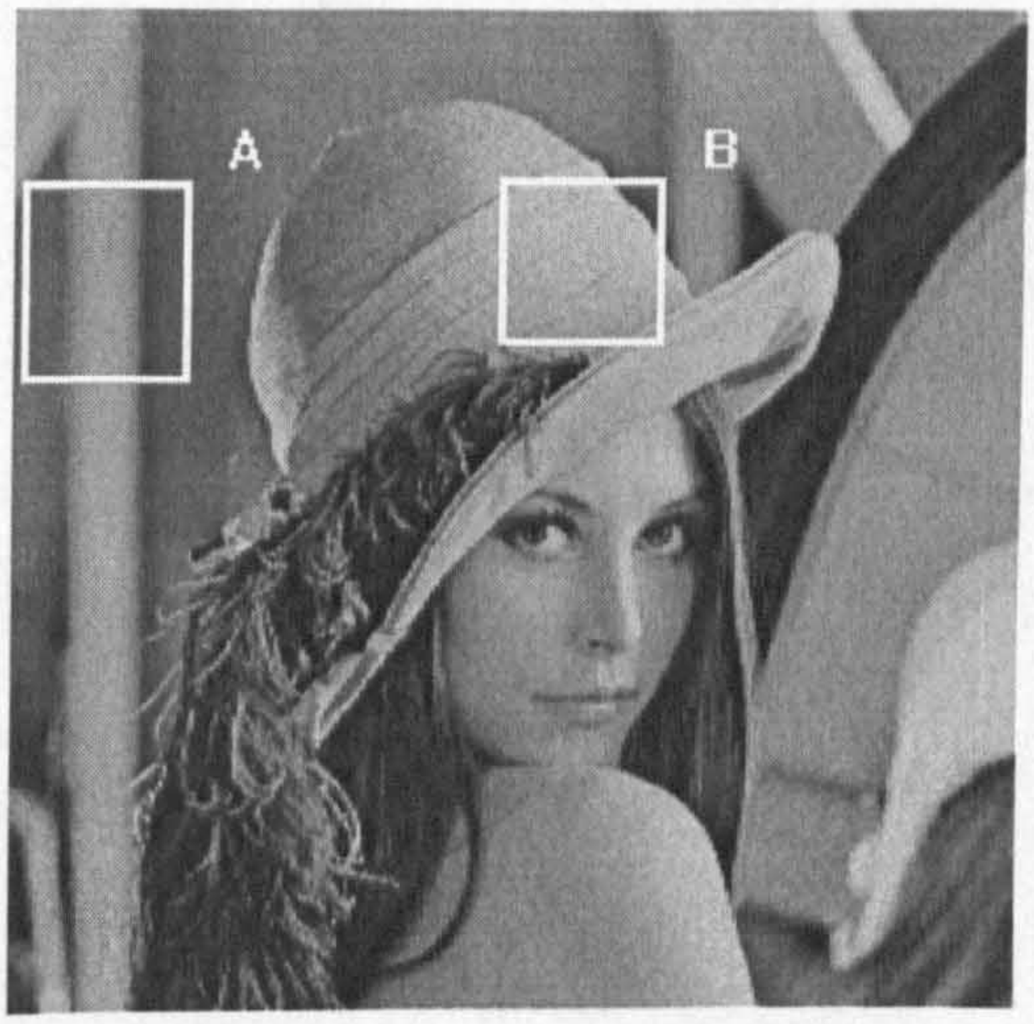

Figure 68:Highlighted details for Lenna image

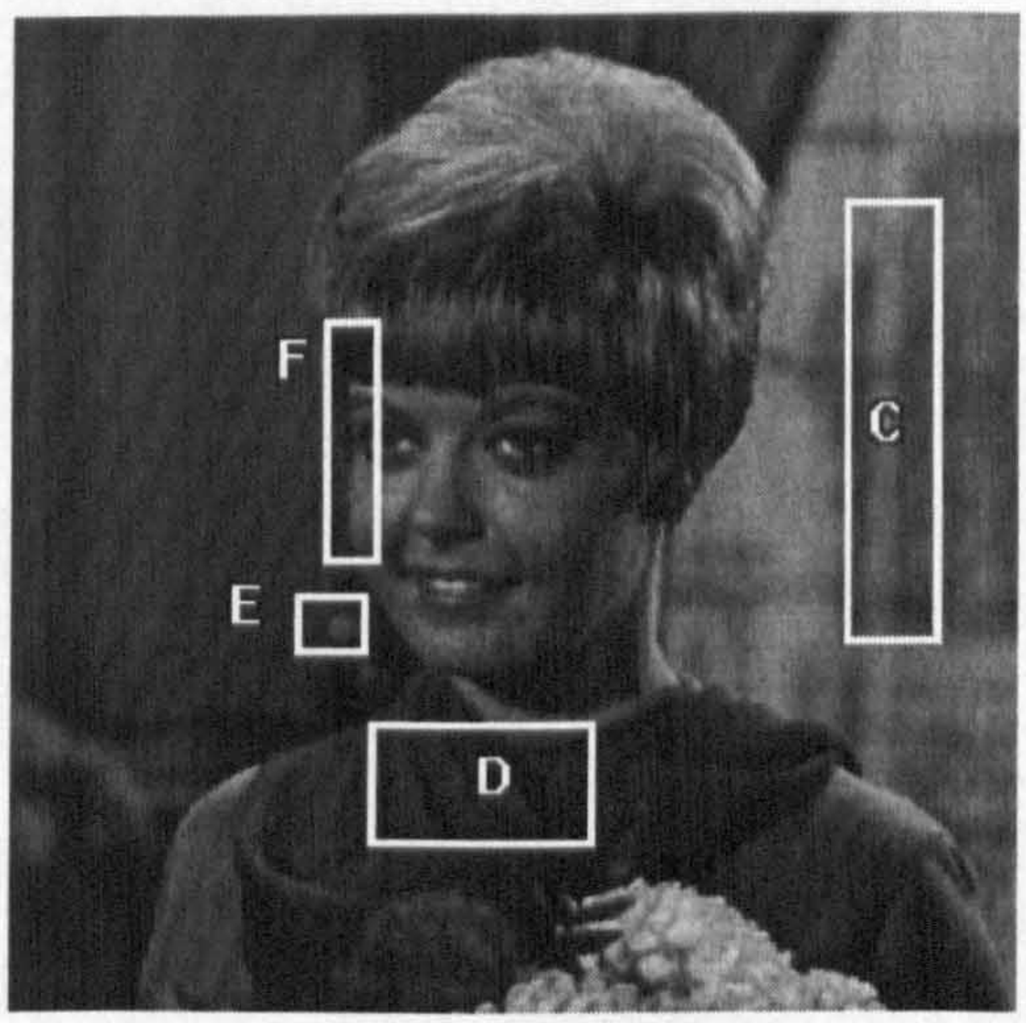

Figure 69:Highlighted details for Girl image

The range of intensities in the edge detection maps leaves a choice as to the threshold value, with a compromise between missed and spurious points to be considered. In the selection of edges it is usually argued that 'the most relevant' edges should be chosen, where these should be the 'higher contrast' edges.

Figure 70 presents detail E for some of the methods tested. The detail is extracted from the produced images, and consists of a matrix of 16x16 pixels. None of the implemented algorithms mark the detail with a uniform grey level. However, this can be achieved by the application of a suitable threshold.

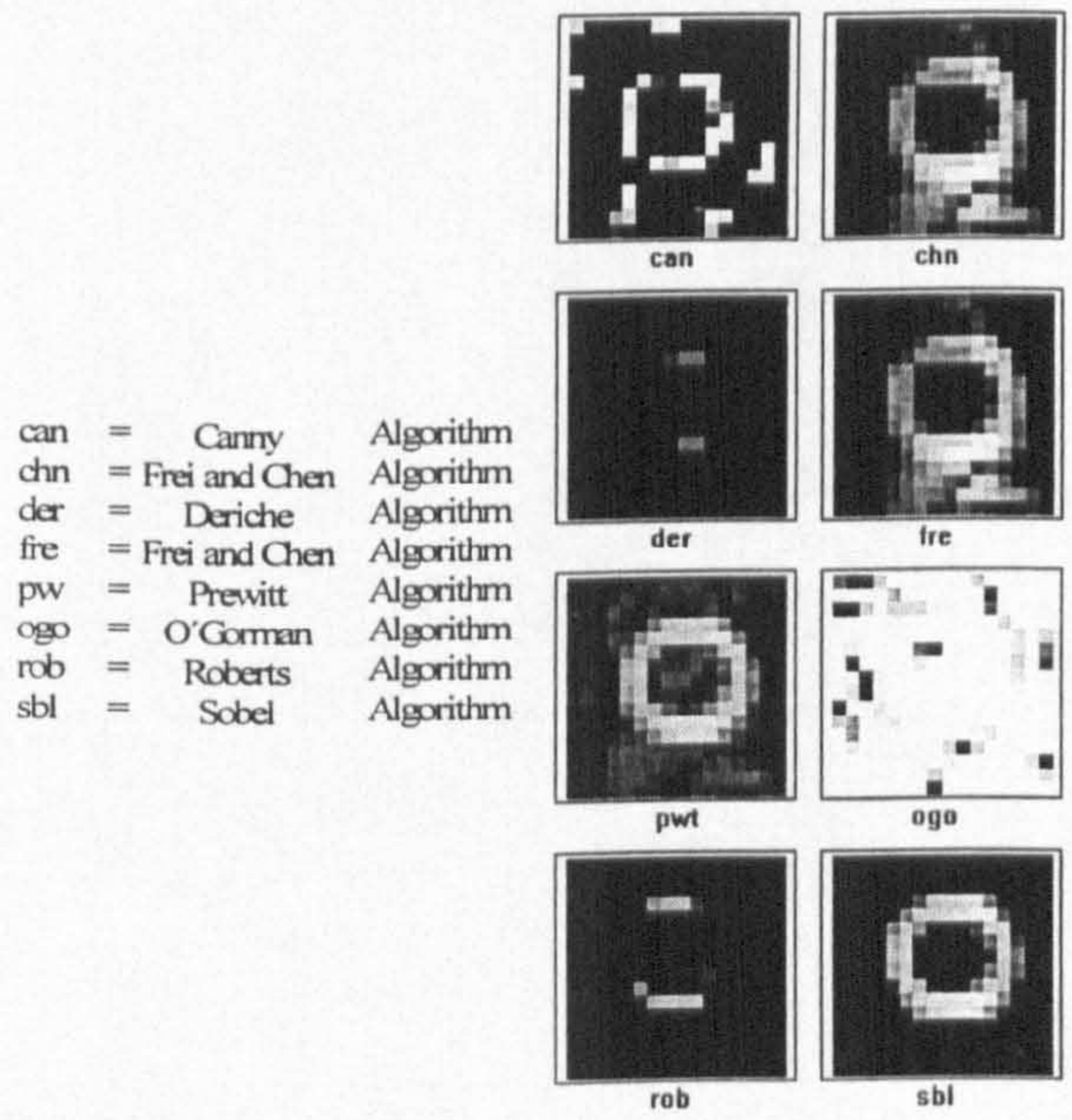

Figure 70: Circles details for the implemented algorithms

The threshold used was based on a visual criteria, as the histogram does not give a clear criteria with these types of maps. The histogram of these images presents a monotone decrease in the number of pixels marked by the different grey levels without giving a clue as to a suitable value for the threshold. The threshold value was chosen in order to obtain the greatest continuity of the edges without including a large amount of points that are originated by the noise in the original picture. In the case of the Sobel and Prewitt operators, a second threshold was used, chosen to obtain images that have a similar number of points marked as the Canny edge map.

TABLE VIII
Number of Marked Points for Girl's Detail E

Method	Threshold	Points marked %	Circle points marked
Roberts	205	7	16
Sobel	219	20	53
	128	8	31
Prewitt	186	20	52
	128	8	32
Frei	200	10	>49*
Canny	255	18	21
Deriche	255	5	13
O'Gormann**	--	--	--

* Unbounded area

** Virtually indistinguishable(almost all points marked)

TABLE VIII presents the count of the number of points that form detail E, the small circle on the left of the girl, together with the value of the threshold and percentage of the points marked in the whole image, for the algorithms considered.

Similar problems arise in the comparison of the squares images. These however, due to the homogeneity of the surfaces, present a smaller number of grey levels in the edge enhanced maps. Squares will be named by row 1 to 8, from top to bottom, and the columns A to H from left to right as shown in Figure 71.

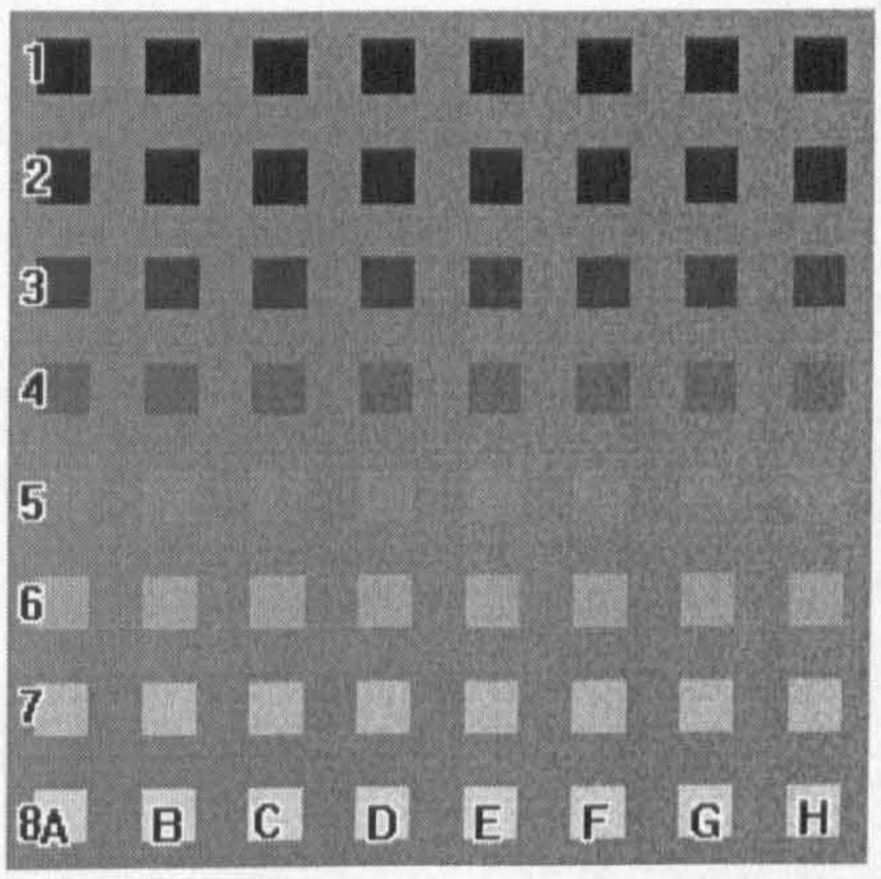

Figure 71:Labelling for squares image

The following table presents in a short form the main features of the squares images:

TABLE IX
Evaluation for Squares Image Edge Maps
(After thresholding)

Method	Line Width	Missed columns (complete)	Missed columns (partial)	Missed rows (complete)	Missed rows (partial)	Square Line Width (Average in Pixels)	Square Line Width (Relative)
Roberts	>1	--	>A	5	4	78.8	1.23
Sobel	>1	--	>B	--	4,5,6	120	1.88
Prewitt	>1	--	>C	--	5	114.45	1.79
Frei	>1	--	>A	5	4	108.8	1.7
Canny	1-I	--	>E	--	5(>E)	60.25	0.94
Deriche	1-R	--	>A	5(?)	4,5	31.28	0.49
O'Gorm.	1/3	--	>C	--	5(>C)	143.98	2.25

Notes: I-irregular
R-regular
> - After ...

A comparison from the squares images presented shows remarkable differences between the different algorithm's robustness to noise handling. These can be seen from the images presented and from TABLE IX which enumerates the main differences.

The differences in the width marked can be seen directly from the edge enhanced maps. However, the threshold of the edge enhanced maps will mark these differences clearly. TABLE IX was rewritten, based on threshold maps, as TABLE X.

TABLE X
Evaluation for Squares Image Edge Maps
(After thresholding)

Method	Thre-shold	marked (%)	missed rows		missed columns	
			complete	partial	complete	partial
Roberts	128	2	5	4	A to C	C to H
	222	2	5	4 to 6	--	all
	238	18	5	1 to 8	--	all
Sobel	128	10	4 to 6	3 to 7	--	all
	177	20	4 to 5	3 to 7	--	all
Prewitt	128	9	4 to 5	--	--	all
	176	10	5	4	--	all
	214	20	5	--	--	all
Frei	128	3	2 to 8	--	--	all
	200	10	4 to 7	1*	--	all
	240	21	5	1 to 3*	--	all
Canny[+]	128	10	all	all	all	all
	219	10	--	5	A.>E	
Deriche	128	3	3 to 7	--	--	all
	234	10	5	4	--	all
	244	19	5	4	--	>B
O'Gormann[¬]	42	10	--	5	--	>D
	84	20	--	5		>E
	128	37	--**	5**	--**	--**

Notes: [+]-Ill defined shape [¬]-Missing corners *-All square marked **- virtually indistinguishable >- After...

The Frei and Chen Operator produces remarkable steady lines. however it does miss some of the low contrast features.

TABLE XI presents measures performed over the squares image for the different thresholds considered. These are written together with the resulting percentage of pixels above the threshold value. The average values of the edges marked are presented as the ratio between the average ($\bar{x}$) marked edge size and the correct edge (16 points) , and the standard deviation (σ) of the values obtained. The average and standard deviation are calculated from the values measured in each 32 x32 sub-image containing one square. Results for the number of marked points inside and outside the squares are also presented.

Zero crossing operators have the advantage that the width of the marked edge is 1 pixel. It is remarkable how low the standard deviation is for the Canny operator when a high threshold is used. Another advantage of the zero crossing operators is that they mark even the poorest contrast features. Low contrast details in the image are clearly marked by the Canny operator, although some effort is necessary to identify them, as there are a number of spurious lines created which relate to the noise handling. This fact is also reflected in the low standard deviations exhibited. These seem to be mainly due to variations in the shape of the squares rather than to variations in the number of marked edges. The Deriche operator produces a stronger separation between strong and weak edges. This fact is reflected by an almost complete row missed by the Deriche operator, even without thresholding, and reflects the highest standard deviation exhibited.

TABLE XI
Square Image Measures

Method	thr	%	Lines		Inside		Outside	
			x̄	σ	x̄	σ	x̄	σ
Roberts	128	2	0.26	0.27	0	0	0	0
	222	2	1.18	0.51	0.02	0.03	0.02	0.03
	238	18	1.42	0.47	0.08	0.1	0.09	0.11
Sobel	128	10	1.29	0.78	0	0.01	0.01	0.01
	177	20	1.63	0.66	0.07	0.09	0.08	0.11
Prewitt	128	9	1.47	0.3	0	0	0	0
	176	10	1.7	0.65	0	0.01	0	0.01
	214	20	1.91	0.53	0.07	0.1	0.1	0.12
Frei	128	3	0.21	0.52	0.1	0.05	0	0
	200	10	1.01	0.95	0.06	0.19	0	0
	240	21	1.87	0.67	0.18	0.28	0.03	0.06
Canny+	128	10	0.5	0.11	0.04	0.03	0.1	0.04
	219	10	0.84	0.07	0.08	0.04	0.2	0.03
Deriche	128	3	0.35	0.45	0	0	0	0
	234	10	0.93	0.32	0.03	0.04	0.03	0.04
	244	19	1.05	0.2	0.1	0.1	0.14	0.12
O'Gormann ++	42	10	0.87	0.16	0.03	0.02	0.06	0.01
	84	20	1.16	0.13	0.11	0.03	0.17	0.02
	128	37	1.77	0.19	0.24	0.04	0.32	0.03

Notes: + - well defined shape; ++ - Missing corners; *thr* - threshold value; % - pixel percentage over *thr*; x̄ - Average; σ - standard deviation

The O'Gormann operator is unique in two aspects. Firstly it marks all points even in the case of very low contrast. Like the Canny operator it produces a good average line with a small standard deviation. Secondly it marks every slope as an edge. For a threshold of 128 the number of points that are not in the line area is larger than any of the other methods. This can be clearly seen in the girl image which is very dark compared to the images produced by the other methods. In the Band A image, the right extreme is missed by the majority of the algorithms (it is even doubtful if an edge exist there). It is however registered by the O'Gormann operator. The strength of the tendency of The O'Gormann operator to mark varying grey level areas with a small

slope can be seen by the average tone in which O'Gormann processed pictures are printed. As in all pictures so far, edges are marked as black points. In O´Gormann maps the white lines produced are effectively the unmarked points.

TABLE XII presents measures obtained over the Band A image. From the Table the different performances of the different edge detectors implemented can clearly be seen, as can their inability to mark perfect edges.

TABLE XII
Band A Edge Map Evaluation

Method	Thr	ratio ()/correct:			EQ	MQ
		Wrong	Correct	Missed		
Roberts	128	0.11	0.08	0.93	0.42	0.07
	222	0.7	0.38	0.62	0.35	0.22
	238	0.94	0.45	0.55	0.32	0.23
Sobel	128	0.52	0.57	0.43	0.52	0.38
	177	0.77	0.77	0.23	0.5	0.44
Prewitt	128	0.69	0.71	0.3	0.51	0.42
	176	0.94	0.85	0.15	0.47	0.44
	214	1.59	0.94	0.58	0.37	0.3
Frei	128	0.05	0.19	0.81	0.79	0.18
	200	0.65	0.59	0.41	0.48	0.36
	240	1.72	0.93	0.07	0.35	0.34
Canny	128	0.82	0.18	0.28	0.18	0.14
	219	0.49	0.3	0.7	0.38	0.2
Deriche	128	0.07	0.05	0.95	0.42	0.05
	234	0.62	0.31	0.69	0.33	0.19
	244	0.79	0.35	0.65	0.31	0.2
O'Gormann	42	0.88	0.31	0.69	0.26	0.16
	84	1.52	0.59	0.41	0.28	0.23
	128	2.01	0.8	0.2	0.28	0.27

Notes: Thr - theshold value EQ - Edge Quality MQ - Map Quality

5.4 Discussion

In this section several edge detection schemes have been described and examples of processed images by these techniques presented. Images produced are the direct output of the edge detection algorithms. before thresholding, or any other decision process. These will be used in subsequent work. As there are many different imaging conditions it is not possible to state that one particular edge detector is optimal for any particular image type.

The presented images show that the proposed goal of neural network arbitration may be achievable. Effectively a more correct, and complete. set of edges could be traced by comparison of the several edge maps. The incomplete edges could be closed by an observer. In some cases this could be done through extrapolating the lines present. In other cases it could be done using knowledge of the objects represented. Neither of these methods is possible using a local approach.

Figure 72 through Figure 83 shows thresholded versions of some of the presented images (Girl. Figure 32 and Lenna. Figure 31). The values in brackets show the percentage of marked points.

Figure 72:Girl image processed by the Roberts operator and thresholded at 205 (7%)

Figure 73: Girl image processed by the Sobel operator and thresholded at 219 (20%)

Figure 74: Girl image processed by the Prewitt operator and thresholded at 186 (20%)

Figure 75: Girl image processed by the Deriche operator and thresholded at 255 (5%)

Figure 76: Girl image processed by the Canny operator and thresholded at 254 (18%)

Figure 77:Girl image processed by the Frei and Chen operator and thresholded at 200

Figure 78: Lenna image processed by the Roberts operator (th=227)

Figure 79: Lenna image processed by the Sobel operator (th=200)

Figure 80: Lenna image processed by the Prewitt operator (th=200)

Figure 81: Lenna image processed by the Frei operator (th=232)

Figure 82: Lenna image processed by the Canny operator (th=254)

Figure 83:Lenna image processed by the O'Gormann operator (th=28)

5.5 Artificial Neural Networks for Edge Detection

5.5.1 Introduction

Within this section a neural network is applied to edge detection. This is carried out for two purposes. Firstly to act as a comparison for the approach to be presented in the next chapter. Secondly as an aid for the strategy development used for the arbitration system to be described in the next chapter. The edge detection case is simpler to test, as the resulting solutions are of smaller dimension than the ones that will be necessary for the arbitration case. Both cases were developed in parallel in order to maintain the compatibility of the systems used. However they will be described separately due to their own particularities.

Multi-layer perceptrons are used throughout the work, as they are one of the most widely used networks [179], exhibit high efficiency, and are good at pattern recognition [119] [161] [184], e.g. edge detection problems [155] [83] [167].

The topology of the neural network is ill-defined and needs to be specified. This is a difficult task as many of the parameters can not be analytically defined. Another core problem is the selection of the learning set. In this section the assumptions made in selecting the network are described.

5.5.2 Topology

A three layer multi-layer perceptron was chosen, since it is the smallest

configuration which can be accepted as an approximation system capable of forming arbitrary decision regions [75]. The number of input parameters is fixed by the size of the windows used. These are used as odd values, to avoid indecision in the position of the edge, as the windows are centred on a pixel. The number of output nodes per decision was fixed to one, corresponding to the classification of edge or non-edge. The number of output neurons could easily be increased if advantage was taken of the parallel implementation. However, it would undoubtedly alter the conditions under which the comparisons could be performed.

5.5.3 Window Size

Another problem is in deciding upon the size of the window. The options as to the size of the window, as isotropy is envisaged, is exactly the same as those used in edge detection filters. This is due to the implementation facility, speed and co-ordinate system used. Square windows were selected, containing an odd number of pixels. These correspond to the smallest integer values (3x3, 5x5, 7x7) as usual for filters. The different windows deal with different amounts of information. It is evident that the larger windows will be slower. However it is expected that, as they deal with more information they will also be more powerful and perform more efficiently than smaller windows.

5.5.4 Parameters

A selection of the learning parameters was initially performed and the chosen

values maintained constant throughout the work. This was done in order to restrict the number of variables in the process. Comparisons are not performed since they are outside the scope of the work. Initial parameters were chosen based on cited values in the literature [119] [145] . Nearby values were tested but did not shown any improvement.

For the learning phase, parameters were fixed as

epsilon	$\varepsilon = 1$
momentum constant	$\alpha = 0.05$
learning constant	$\eta = 0.9$

TABLE XIII

5.5.5 Learning

Several hypothesis for the definition of a learning set were considered. The simplest used an artificially generated set of vectors that correspond to what was traditionally understood as an edge. The second is to take a number of sample images, which contain more complex patterns, and take the training set from these. For these training sets two methods for obtaining the data from the training images were considered. The next two sections deal with the two different training sets and the methods for obtaining the data.

5.5.5.1 Training data through vector generation

Several hypothesis for the definition of a learning set were considered. The

first one, the simplest, is to generate artificially a set of vectors that correspond to standard situations that are unequivocally understood as edges or non-edges. These allow a controlled distribution of the edges, but hardly represent all the types of edges that will be present in a real image. Edges were generated as step edges, with different amplitudes and different orientations.

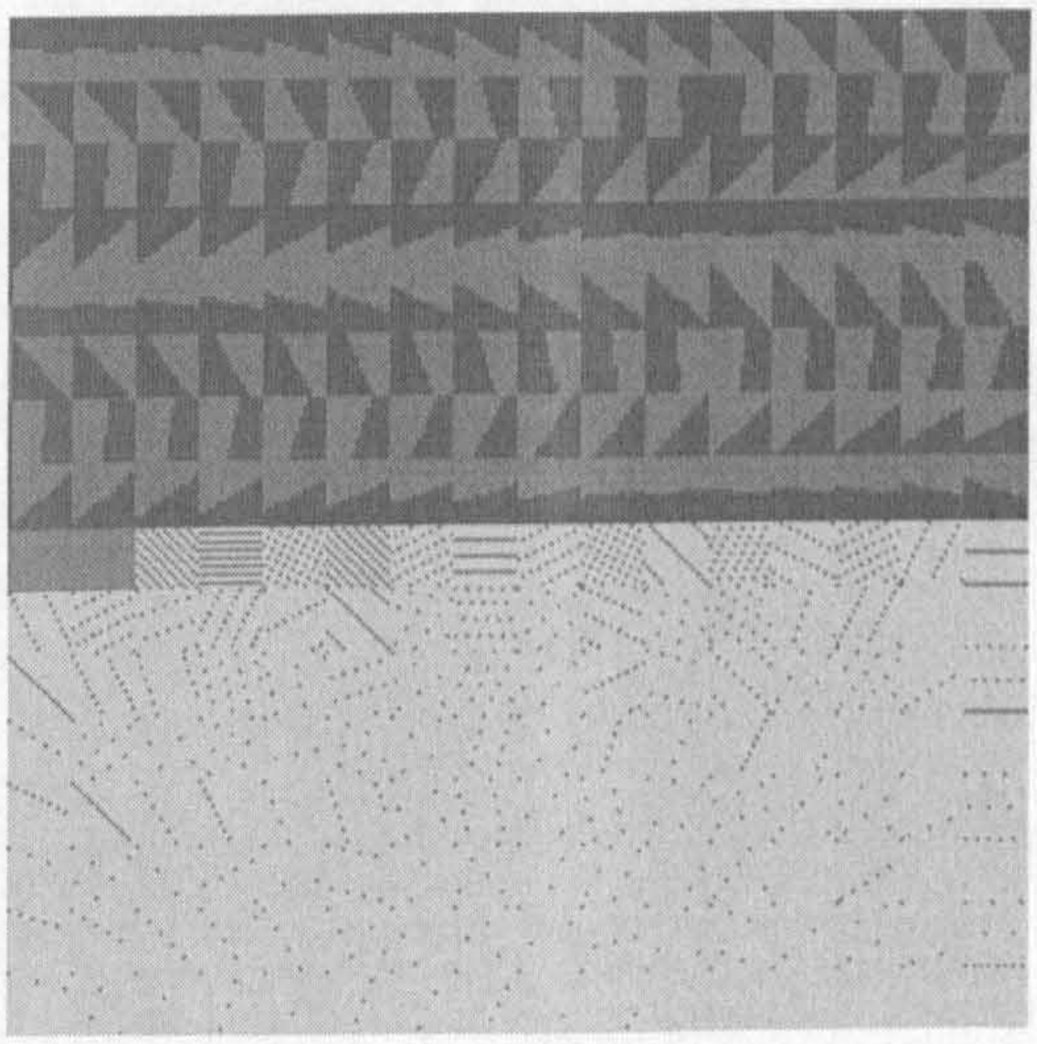

Figure 84 : Generated learning set

Examples of such sets are represented in Figure 84 . The upper half constitute the edges subset, where a step edge is generated and rotated over itself by 360 degrees. This figure could then be scaled to give different step amplitudes. The lower half, which is the non-edge subset, is composed of points, having different spacing (the lines are simply aliases), which could also be scaled to cover different intensities.

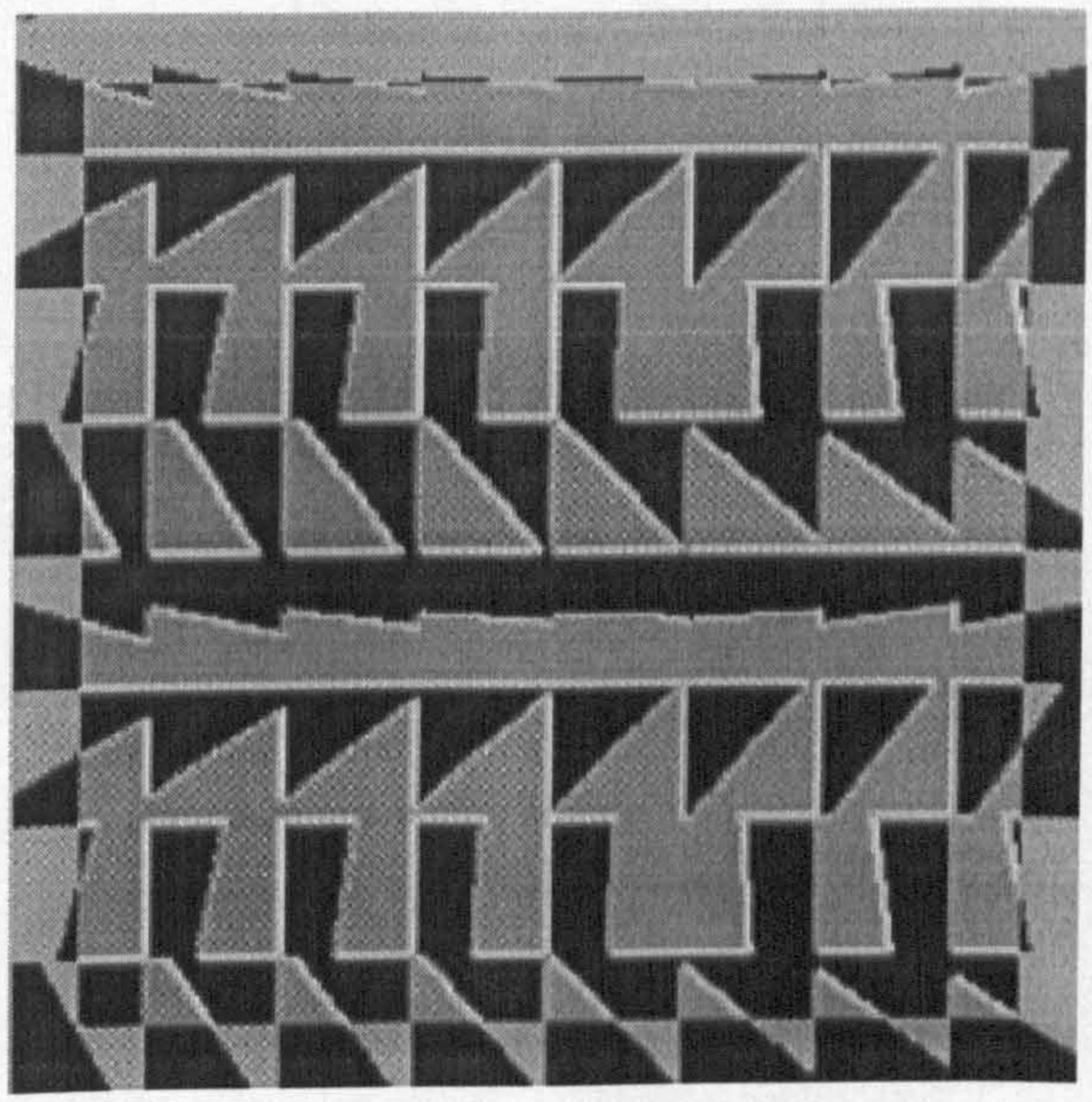

Figure 85: An example result from the first networks tested (5^2x5x1)

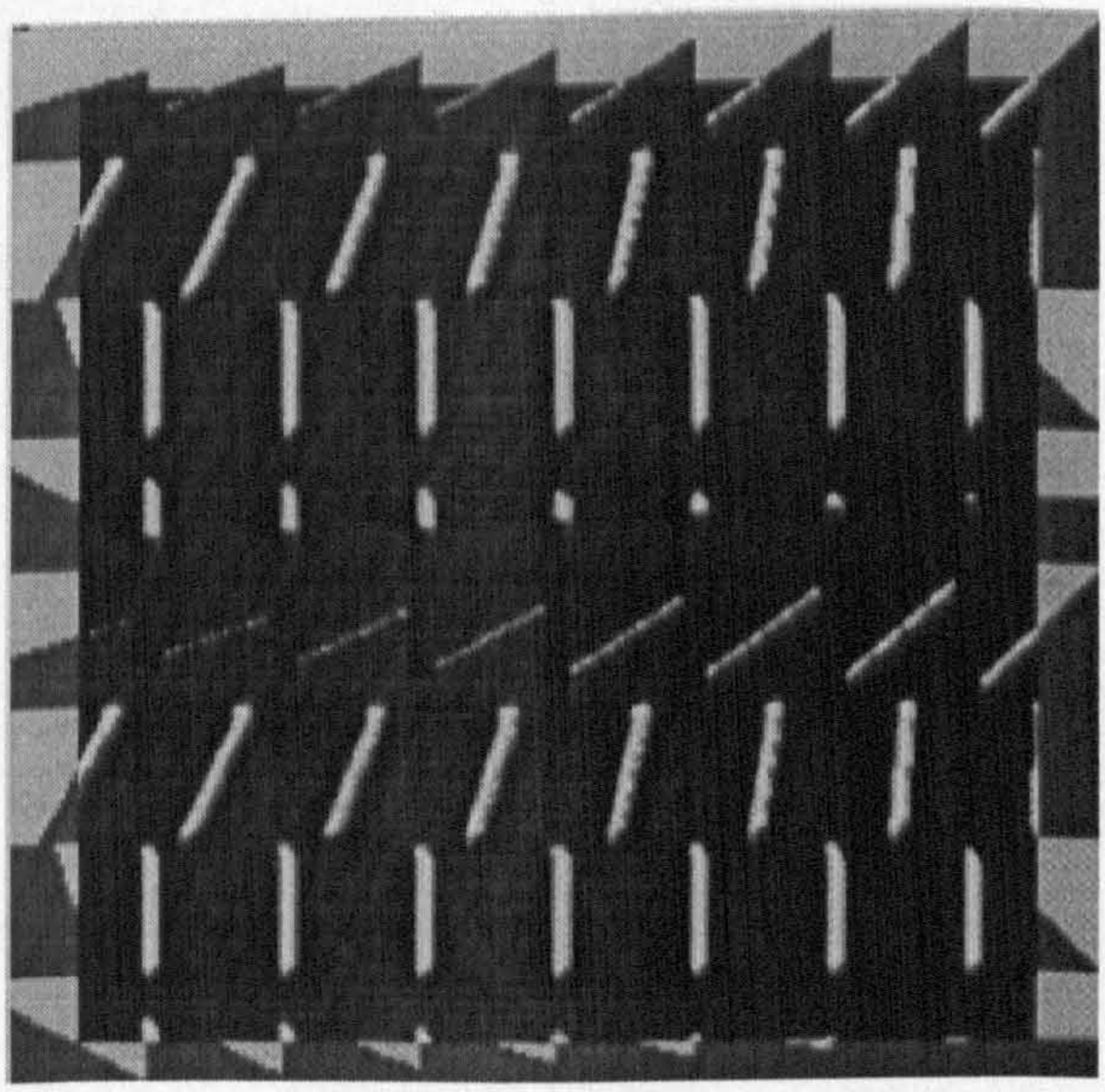

Figure 86: An example result from the first networks tested (72x8x1)

From this picture windows of several sizes can be extracted, as the positions are known, and the position will contain the information of which patterns

could be interpreted as edges or non-edges. Two examples of the performance of such a networks are presented in Figure 85 and Figure 86. In these cases the networks were a 5^2x5x1 and a 7^2x8x1 multi-layer perceptron type

Networks trained with such a training set, as shown in figure 84, presented poor performance. Not only are the marked edges quite thick, but also the response varies according to the grey level. This system could be improved through the inclusion in the edge model of more information to force the solution to shrink the edge size, such as non centred edge patterns.

5.5.5.2 Training data through image scanning

A more elaborate solution is to generate images, containing more complex patterns, from which the patterns can be collected. This can be done using sample images. Image processing systems are normally applied to similar images which contain common characteristics, not only due to the objects, but also due to the distortion repetitiveness of the acquisition system. However, this solution has some disadvantages during the development stage. Correct edges are sometimes difficult to trace, due to the blurring from the acquisition device. This procedure will require the use of a screen pen or similar device, which allows for more movement co-ordination than that allowed by a mouse for drawing, to input the edge positions. Due to image blur edge tracing turns out to be a difficult task, as the shape and position is lost when small areas of the image are zoomed in on. A mixed solution was implemented, where from generated images samples were collected and used both for the learning phase and for the evaluation of the solution obtained. The process for collecting the

training data is shown in Figure 87

Sample images are shown in Figure 88 and Figure 90 , with reference edges traced in Figure 89 and Figure 91 respectively. A full image or a reduced version of the image, is scanned, from left to right and from top to bottom to select the area of interest.

Figure 87:Process for collecting data

The area of interest is then scanned a second time to extract the patterns, which are stored in a file. This file contains vectors which correspond to edge and non-edge patterns.

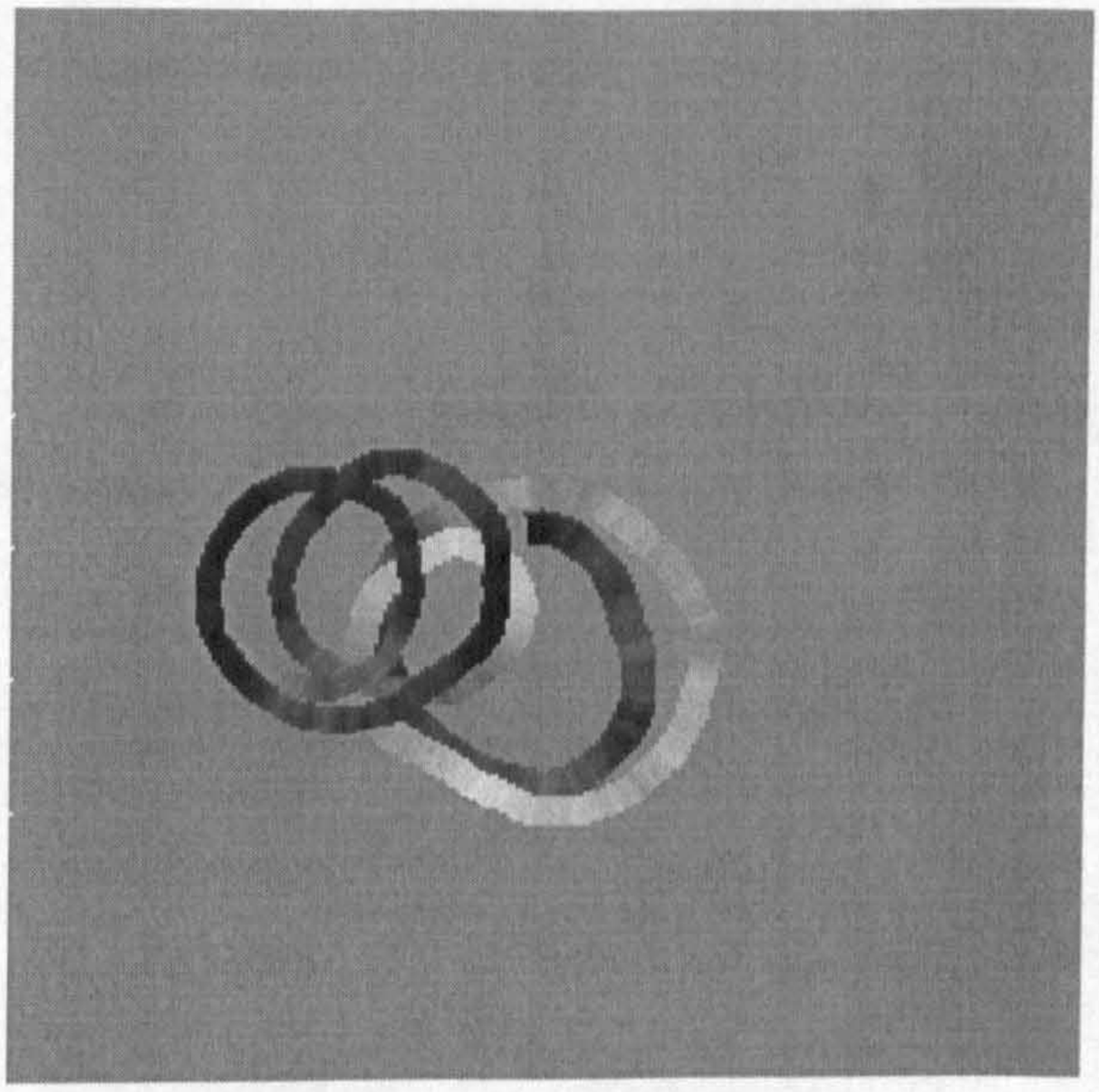

Figure 88: Band A

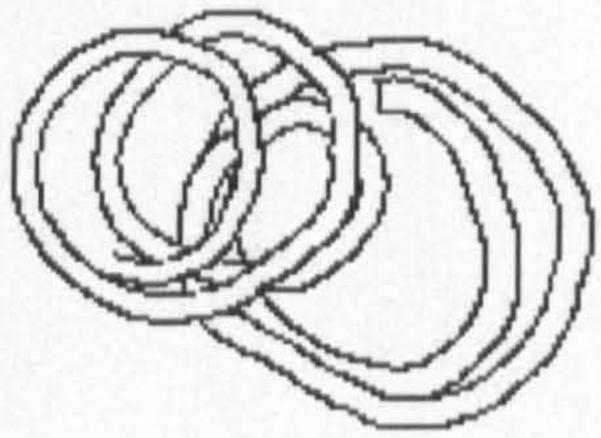

Figure 89: Band A - Reference Edges

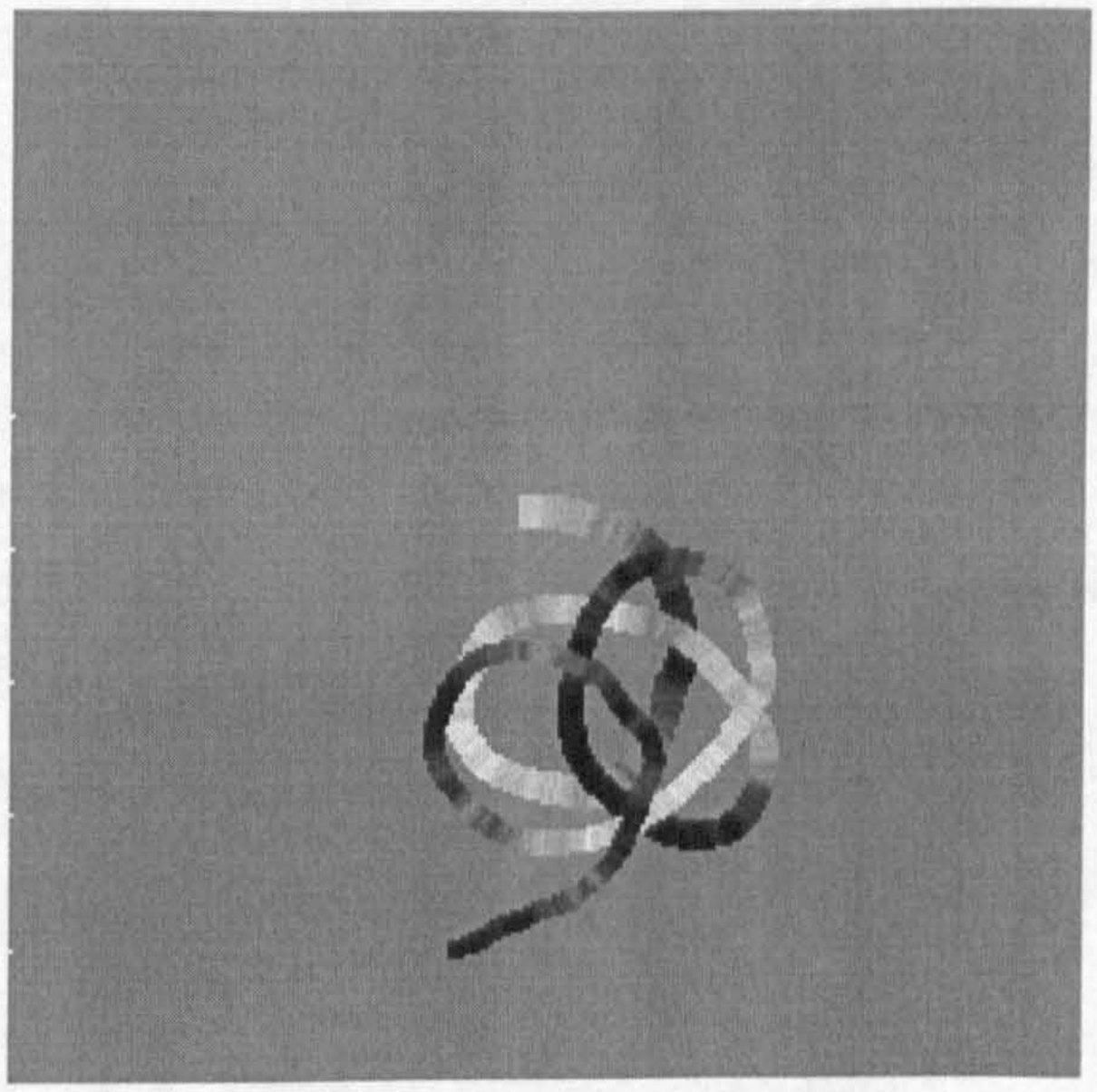

Figure 90 :Band B

Figure 91: Band B - Reference edges

These sets were reduced through the use of sequential sampling of the images. The image generation process is not sequential and thus patterns collected

sequentially appear as virtually random. This procedure decreases the problem size and thus allows for smaller learning times. However, the trained networks presented a poor performance. Edges marked by the networks, taught using reduced learning sets, appear in some areas as regularly spaced dots, which clearly reflect the spacing from the learning set. These gaps were successively reduced, and finally eliminated when the full reduced image was used. The networks referred to throughout this chapter operated on a reduced image (128x128) and without any form of sampling.

Several networks were taught with the image presented in Figure 90 . TABLE XIV presents the learning rates obtained by these networks. The size of the learning set used is presented for each hidden node number tried, the recalling factor achieved and the iteration at which it was obtained.

The neural networks. which will be referred to in the remainder of this chapter. were taught with complete sets of patterns extracted from reduced versions of the images shown in Figure 88 and Figure 90. The obtained networks were then tested on images which are similar in nature, (Figure 88 or Figure 90) and on images that are not similar (Figure 31: Lenna. Figure 32 :Girl, and Figure 33: Squares).

The processed images are presented prior to the hard limiter. This allows for the assessment of the real performance of the network. Measures presented are carried out for thresholded images at 128. as this corresponds to the middle value of the sigmoid used.

TABLE XIV
Final values obtained with a full learning set extracted from a reduced Band B image

Window Size	Final Values			Learning Set Size	
3x3				2,255	
HidNod	WP	RCD	%	@	RMS
10	114	2,141	94.94	5,000	0.05
20	75	2,180	96.67	5,000	0.03
30	65	2,190	97.12	5,000	0.03
40	80	2,175	96.45	5,450	0.03
50	73	2,182	96.76	5,000	0.03

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XIV - A

Window Size	Final Values			Learning Set Size	
5x5				2,255	
Hid Nod	WP	RCD	%	@	RMS
10	63	2,192	97.21	3,050	0.03
25	31	2,224	98.63	3,500	0.01
30	21	2,234	99.07	5,050	0,01
40	35	2,220	98.45	6,450	0.02
50	41	2,214	98.18	1,700	0.02

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XIV -B

Window Size	Final Values			Learning Set Size	
7x7				2,255	
Hid Nod	WP	RCD	%	@	RMS
10	40	2,215	98.23	5.000	0.02
15	33	2,222	98.54	5.000	0.01
25	22	2,233	99.02	5.000	0.01
30	30	2,225	98.67	5.000	0.01
40	25	2,230	98.89	2,450	0.01
50	18	2,237	99.2	5.000	0.01
60	15	2,240	99.33	5.000	0.01
70	18	2,237	99.2	5.000	0.01
90	18	2,237	99.2	5.000	0.01

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XIV - C

TABLE XV
Performance of neural network edge detectors on the Band images after training on a full training set of Band B

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3x3	10	0.86	0.23	0.95	0.27
	20	0.85	0.31	0.89	0.33
	30	0.82	0.29	0.9	0.35
	40	0.87	0.34	0.9	0.36
	50	0.86	0.26	0.9	0.31

EQ -Edge Quality MQ - Map Quality

TABLE XV - A

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5x5	10	0.49	0.27	0.53	0.27
	25	0.74	0.33	0.77	0.36
	30	0.5	0.31	0.59	0.35
	40	0.62	0.34	0.72	0.37
	50	0.56	0.34	0.7	0.37

EQ -Edge Quality MQ - Map Quality

TABLE XV - B

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
7x7	10	0.57	0.21	0.61	0.26
	15	0.43	0.25	0.49	0.28
	25	0.56	0.27	0.63	0.28
	30	0.49	0.22	0.66	0.56
	40	0.5	0.29	0.61	0.32
	50	0.35	0.22	0.48	0.28
	60	0.57	0.3	0.58	0.3
	70	0.55	0.31	0.52	0.31
	90	0.44	0.28	0.56	0.31

EQ -Edge Quality MQ - Map Quality

TABLE XV -A

TABLE XVI
Performance of neural network edge detectors on the Squares Image
after training on a full training set of Band B

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3x3	10	0.31	0.24	0	0	0	0
	20	2.61	0.62	0.51	0.3	0.92	0.02
	30	0.41	0.29	0	0	0.01	0.01
	40	0.4	0.27	0.01	0.02	0	0
	50	0.29	0.24	0.01	0.03	0.01	0.02

$\bar{x}$ - Average value σ - Standard deviation

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
5x5	10	0.31	0.28	0.14	0.23	0	0
	25	0.38	0.18	0.02	0.07	0	0
	30	0.51	0.26	0.24	0.23	0.01	0.01
	40	0.38	0.24	0.08	0.02	0	0
	50	0.4	0.19	0.13	0.22	0	0

$\bar{x}$ - Average value σ - Standard deviation

TABLE XVI - B

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
7x7	10	0.3	0.27	0.14	0.24	0	0
	15	0.45	0.27	0.17	0.19	0.01	0.01
	25	0.44	0.31	0.12	0.18	0	0
	30	0.51	0.27	0.23	0.23	0.01	0.02
	40	0.36	0.29	0.17	0.02	0	0
	50	0.46	0.25	0.38	0.2	0	0
	60	0.43	0.24	0.08	0.09	0	0
	70	0.56	0.31	0.27	0.2	0	0
	90	0.53	0.29	0.21	0.22	0	0

$\bar{x}$ - Average value σ - Standard deviation

TABLE XVI - C

The results obtained for the bands images are better than those using the conventional edge detection schemes with values of MQ in the order of 25%, presented in TABLE XII, for the edge detection schemes whilst values in the range of 27% to 37% were obtained for the neural network detectors. Differences for the EQ figure of merit are more impressive, with values in the order of 40% for the edge detection schemes whilst values in the range of 50% to 90% were obtained for the neural network detectors

5.5.5.3 Training data through reduced image scanning

In the case of the previous learning sets the vectors that contain only pixels from the background turn out to be larger than the number of other patterns. As this could affect the learning or make the learning process long, a second set of networks were taught. In this case windows which only contain the background points are discarded, with the exception of the first window as at least one background point is important to the learning process. When this value was missed some networks marked areas with a constant grey level as edges. This behaviour can still be observed in some of the solutions presented. Several networks were investigated, corresponding to different network hidden layer sizes and different learning vectors. The size of the learning sets used are shown in the final value tables. It is important to note that, as the mask size increases, the number of patterns that 'touch' an edge pattern increases. The increase of the learning sets does not mean that more edges exist. It simply means that there are more possible non-edge patterns and thus the relation between the edge and non-edge patterns decreases with the size of the mask.

Final values for the more representative networks taught with the reduced learning set are presented in TABLE XVII. TABLE XVIII and TABLE XIX which presents their performance over the bands and squares images respectively.

TABLE XVII
Final values obtained with a reduced learning set
extracted from a reduced Band B image

Window Size	Final Values			Learning Set Size	
3x3				1,412	
HidNod	WP	RCD	%	@	RMS
10	131	1,281	90.72	3,250	0.09
15	120	1,292	91.5	5,000	0.07
20	96	1,316	93.2	3,250	0.07

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XVII - A

Window Size	Final Values			Learning Set Size	
5x5				1,702	
HidNod	WP	RCD	%	@	RMS
20	35	1,667	97.94	5,000	0.02
25	35	1,667	97.94	1,850	0.02
30	21	1,681	98.77	6,650	0.01

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XVII - B

Window Size	Final Values			Learning Set Size	
7x7				1,811	
HidNod	WP	RCD	%	@	RMS
10	30	1,781	98.34	2,500	0.02
30	16	1,795	99.12	2,500	0.01

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XVIII
Performance of the neural network edge detector for the Bands Image after training on a reduced training set of Band B

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3x3	10	0.88	0.21	0.93	0.25
	15	0.85	0.27	0.9	0.31
	20	0.88	0.28	0.89	0.33

EQ -Edge Quality MQ - Map Quality

TABLE XVIII - A

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5x5	20	0.39	0.24	0.47	0.28
	25	0.61	0.27	0.69	0.32
	30	0.42	0.27	0.5	0.31

EQ -Edge Quality MQ - Map Quality

TABLE XVIII -B

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
7	10	0.57	0.21	0.61	0.26
	30	0.57	0.24	0.61	0.3

EQ -Edge Quality MQ - Map Quality

TABLE XVIII - C

TABLE XIX
Performance of the neural network edge detectors on the Squares Image after training on a reduced training set of Band B

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		ℵ	σ	ℵ	σ	ℵ	σ
3x3	10	0.22	0.25	0	0	0	0
	15	0.33	0.27	0.01	0.04	0	0
	20	0.41	0.3	0.03	0.08	0	0.01

ℵ - Average value σ - Standard deviation

TABLE XIX - A

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		ℵ	σ	ℵ	σ	ℵ	σ
5x5	20	0.34	0.19	0.2	0.24	0.01	0.01
	25	0.38	0.21	0.03	0.05	0	0.01
	30	0.47	0.26	0.29	0.23	0	0.01

ℵ - Average value σ - Standard deviation

TABLE XIX - B

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		ℵ	σ	ℵ	σ	ℵ	σ
7x7	10	0.33	0.23	0.04	0.06	0	0.01
	30	0.31	0.2	0.09	0.16	0	0

ℵ - Average value σ - Standard deviation

TABLE XIX - C

5.5.5.4 Convergence criteria

One traditional problem. during the learning phase. is the decision of when to stop teaching. The size of the learning set does not allow for the inclusion of a control set due to the lack of memory. Thus it is not possible to evaluate the capabilities of the actual solution on a different set. The alternative would be

to constantly use the hard disk as auxiliary memory. This would make the learning process extremely slow. In the first instance the program was developed on a single transputer TRAM situated on a TMB16 board within a PC. This was useful for the development on small networks. However for larger networks the much larger learning times made the system impractical. This prompted the program to be transferred to the main University VAX system. Even here. some of the solutions presented learning times that gave problems with the use of batch queues. Some of the solutions presented were taught in successive runs on-line over several nights.

The mean square error is monitored during the learning phase. The fact that this error decreases during the learning phase does not mean that a better solution was being achieved. After an initial strong learning phase, the mean square error gets smaller and smaller without any decrement in the number of wrongly classified patterns. In fact. after a certain number of iterations. the mean square error does not reflect the number of patterns learnt. This is due to the fact that as the output values get successively closer to the interval extremes the measured error decreases independently of an improvement in the learning rate. Figure 92 through Figure 94 show the evolution of the number of unlearnt patterns in three of the networks.

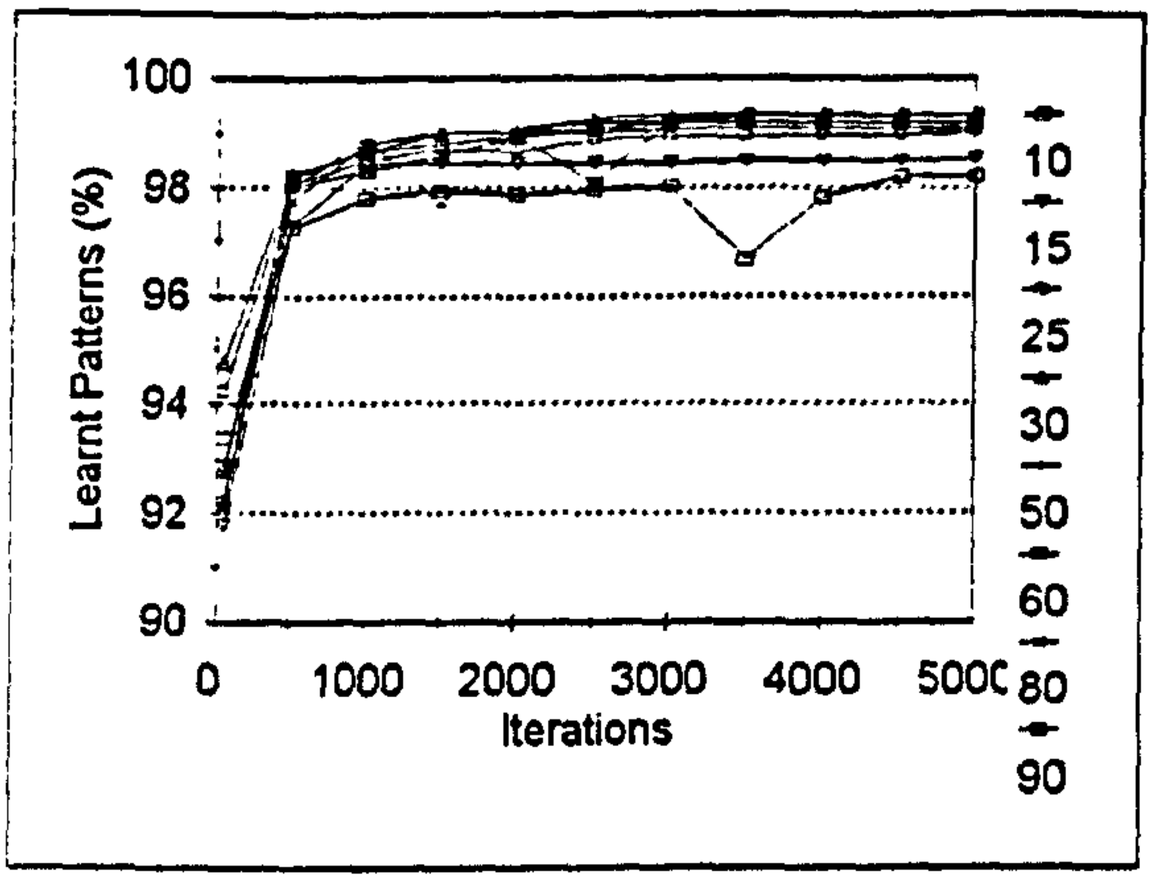

Figure 92:History of patterns learnt for several different number of hidden nodes for a 7^2 input network

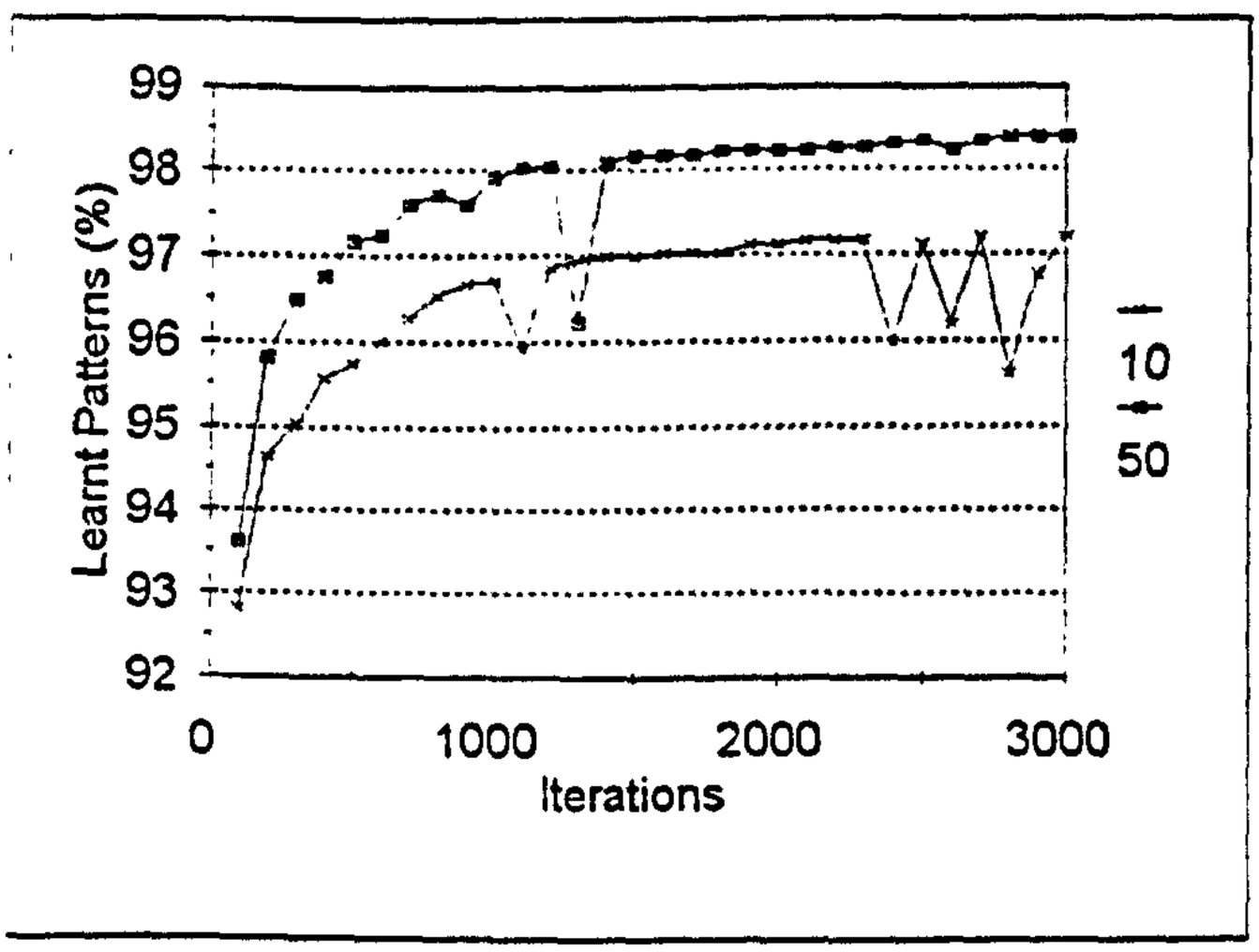

Figure 93: History of patterns learnt for two different number of hidden nodes for a 5^2 input network

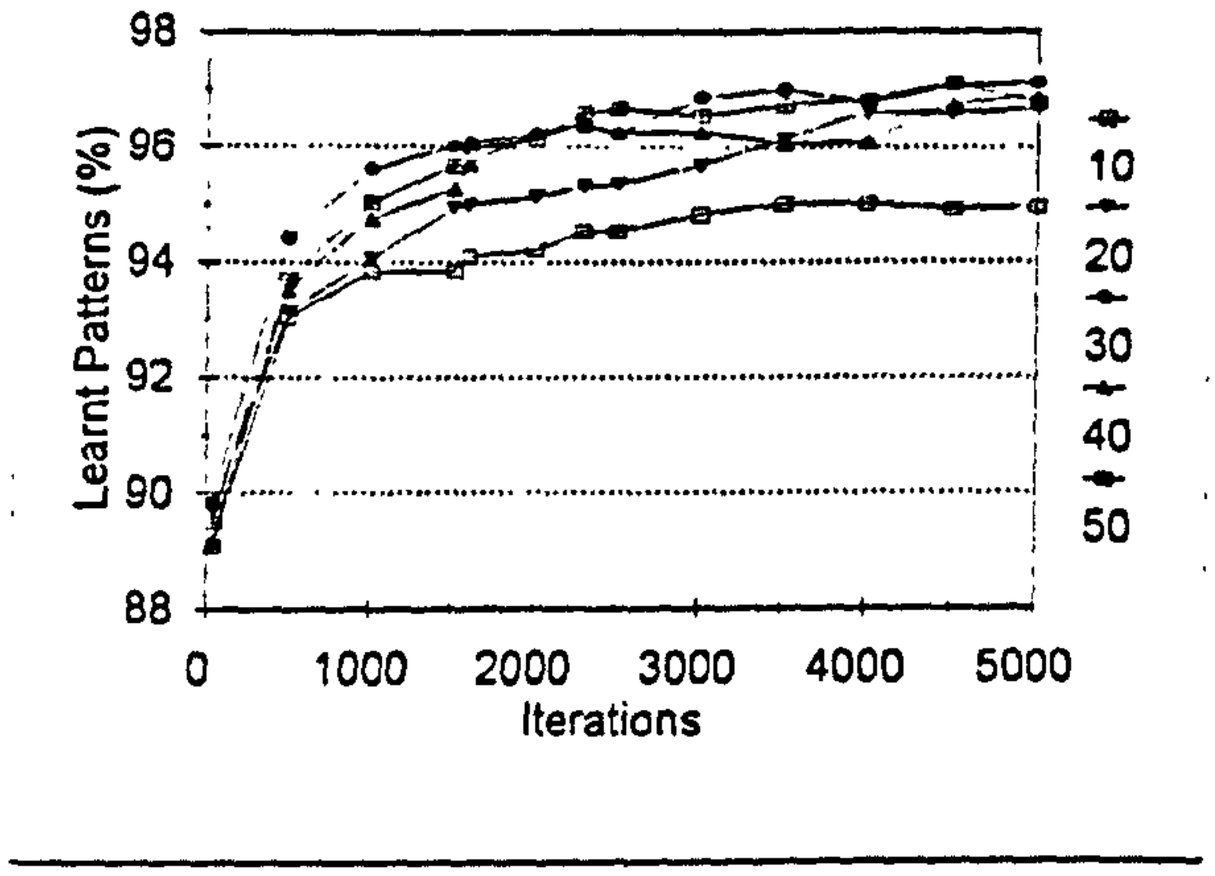

Figure 94:History of patterns learnt for several different number of hidden nodes for 3^2 input network

Figure 93 corresponds to a network with 5^2 input nodes, (i.e. 5x5 squared window), for up to 3 thousand iterations. The learning set was extracted from the picture in Figure 90 . and consists of 2255 points. Figure 94 corresponds to a network with 3^2 input nodes for up to 5000 iterations. After an initial phase the learning rate successively fades. After a certain number of iterations the number of patterns actually learnt does not improve significantly, although a more robust solution is achieved. However, in a few cases, a continuously improving solution was registered. One example is present in Figure 94. This behaviour, however, is not even typical of a particular window size.

A common criteria used when deciding to stop the learning phase is to allow the network to learn until 90 to 95 % of the patterns have been learnt. In the previous figure it can be seen that after 50 iterations the network has reached a 90% success rate (232 wrong patterns out of 2255). In each of the learning sets, non-edge patterns roughly count for at least 2/3 of the points, which

means that the learning phase starts with approximately 66% of the patterns learnt.

In the most equilibrate case (3 pixels size), for each edge pattern there will be 2 non-edge patterns, as Figure 95 illustrates:

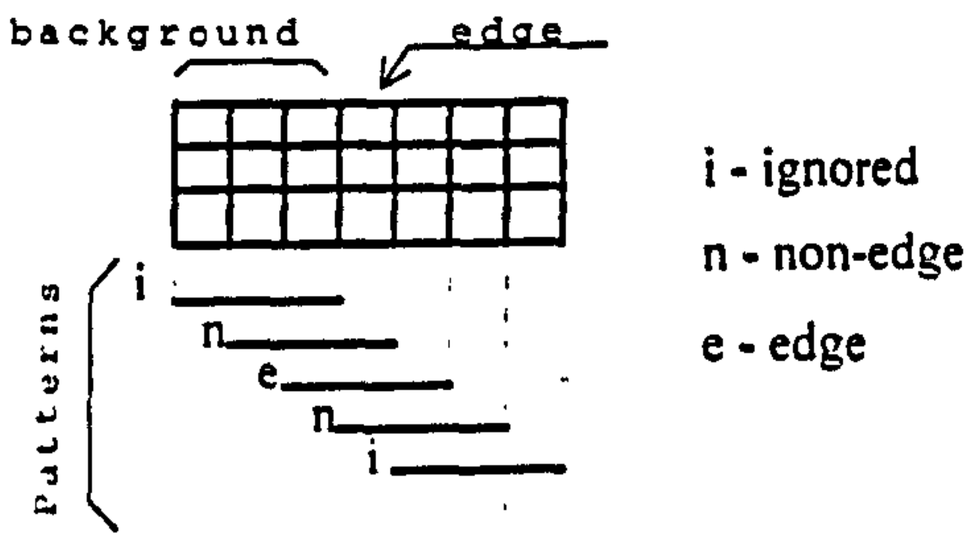

Figure 95: Pattern extraction

In the majority of cases the network was left running until a stable solution was achieved.

To ensure that differences in results are due only to the different number of iterations two of the first networks. trained with a reduced learning set and presented in TABLE XVII were taught a second time with the number of iterations limited to 1000. TABLE XX and TABLE XXI presents the performances for the bands and squares images respectively obtained by these networks.

TABLE XX
Performance of neural network edge detectors on the Band images after training on a reduced training set of Band B with a limited number of 1000 iterations

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3x3	5	0.94	0.23	0.87	0.21
	20	0.88	0.26	0.88	0.25

EQ -Edge Quality MQ - Map Quality

TABLE XXI
Performance of neural network edge detectors on the Squares image after training on a reduced training set of Band B with a limited number of 1000 iterations

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3x3	5	0.07	0.08	0	0	0	0
	20	0.28	0.26	0.02	0.07	0	0

$\bar{x}$ - Average value σ - Standard deviation

Figure 96 and Figure 97 show the Lenna image processed by a 3^2x20x1 network, using the same learning set but with a different number of learning iterations. Generally there are no significant differences between the images. Some of the lines are present in only one of them., e.g. the column on the left side and the hair line on the right side.

However. the higher contrast in Figure 96 is clearly visible. This is due to the large binarisation of the network response. However marked lines are dotted when the extended learning has been carried out. This could be due to over learning. When over-learning occurs the space partition is done locally to the points in the learning set. thus reducing the number of points that are included

in that partition.

Figure 96:Lenna image processed by a 3^2x20x1 neural net after being trained with a full learning set for 5000 iterations

Figure 97: Lenna image processed by a 3^2x20x1 neural net after being trained with a full learning set for 1000 iterations

TABLE XXII
Final values of the neural network edge detectors on the Band Images after training on a reduced training set of Band A

Window Size	Final Values			Learning Set	
3x3				1,811	
HidNod	WP	RCD	%	@	RMS
5	207	1,604	88.57	1,050	0.09
10	179	1,632	90.12	1,050	0.08
15	142	1,669	92.16	1,050	0.06
20	140	1,671	92.27	1,050	0.07

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXII - A

Window Size	Final Values			Learning Set	
5x5				2,170	
HidNod	WP	RCD	%	@	RMS
15	62	2,108	97.14	1,050	0.03
20	68	2,102	96.87	1,050	0.03
25	63	2,107	97.1	1,050	0.03

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXII - B

Window size	Final Values			Learning Set	
7x7				2,386	
HidNod	WP	RCD	%	@	RMS
20	26	2,360	98.91	2,500	0.01
30	45	2,341	98.11	1,150	0.02
40	25	2,361	98.95	2,000	0.01
50	35	2,351	98.53	2,100	0.01
60	47	2,339	98.03	500	0.02

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXII - C

TABLE XXIII
Performance of the neural network edge detectors on the Band Images after training on a reduced training set of Band A

Window Size.	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3	5	0.92	0.31	0.91	0.32
	10	0.96	0.31	0.66	0.9
	15	0.94	0.43	0.62	0.82
	20	0.96	0.35	0.64	0.86

EQ -Edge Quality MQ - Map Quality

TABLE XXIII - A

Window Size.	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5	15	0.54	0.36	0.51	0.36
	20	0.85	0.37	0.73	0.34
	25	0.65	0.37	0.62	0.36

EQ -Edge Quality MQ - Map Quality

TABLE XXIII -B

Window Size.	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
7x7	20	0.66	0.32	0.58	0.31
	30	0.65	0.34	0.58	0.3
	40	0.55	0.36	0.51	0.34
	50	0.76	0.41	0.56	0.32
	60	0.49	0.33	0.5	0.32

EQ -Edge Quality MQ - Map Quality

TABLE XXIII -B

TABLE XXIV
Performance of the neural network edge detectors on the Squares Image after training on a reduced training set of Band A

Window Size.	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3x3	5	0.17	0.17	0	0.01	0	0
	10	0.23	0.22	0	0.02	0	0
	15	0.34	0.21	0.02	0.05	0.01	0.01
	20	0.23	0.17	0.01	0.02	0	0

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXIV - A

Window Size.	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
5x5	15	0.63	0.37	0.23	0.24	0.02	0.02
	20	0.4	0.34	0.02	0.04	0.01	0.01
	25	0.48	0.27	0.13	0.16	0.01	0.01

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXIV - B

Window Size.	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
7x7	20	0.4	0.29	0.09	0.16	0	0
	30	0.43	0.29	0.06	0.07	0	0
	40	0.57	0.39	0.25	0.22	0.01	0.02
	50	0.51	0.33	0.12	0.15	0.01	0.01
	60	0.58	0.3	0.2	0.15	0.05	0.05

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXIV - C

5.5.5.5 Learning set origin

Another problem is from where to extract the learning set. Two band images have been presented within this thesis. Up to now the Band B image has been used for the training of the neural network. To evaluate the best source. another set of networks were taught with learning sets extracted from Band A

image. TABLE XXII presents the final values for the networks referred to . Their performance on the bands and squares image are presented in TABLE XXIII and TABLE XXIV respectively.

The learning sets that can be extracted are larger in this case than in the case of the band B image. Networks trained on the band A image exhibit a higher number of unlearnt patterns and slightly worse relative recalling.

To evaluate the difference between the solutions the Band B image was processed by networks of the same size, one trained on the Band A image, the other trained on the Band B image. The processed pictures were first thresholded, and then blended. The blended image was enhanced afterwards with false colours and is presented in Figure 98. Common marked points are shown in red, which are the majority. Points where disagreement exists are shown in blue and green. The biggest difference resides in the fact that one of the solutions (in blue) marks more points inside the band than the other. However. no significant difference exists and in both cases the marked common points are the same.

The number of points were counted and it was found that there were more blue points than green. This is still true if the marked areas inside the band are cleaned by hand. This could suggest that the image in Figure 90 is more suitable for producing training sets than the image in Figure 88. This can be seen if we compare Figure 99 with Figure 97 where less edges are marked in the later. e.g. the left side vertical bar. albeit more defined and continuous. However this result can not be generalised. as the comparison of Figure 99 with Figure 96 suggests.

with Figure 96 that less edges are marked, e.g. the left side vertical bar.

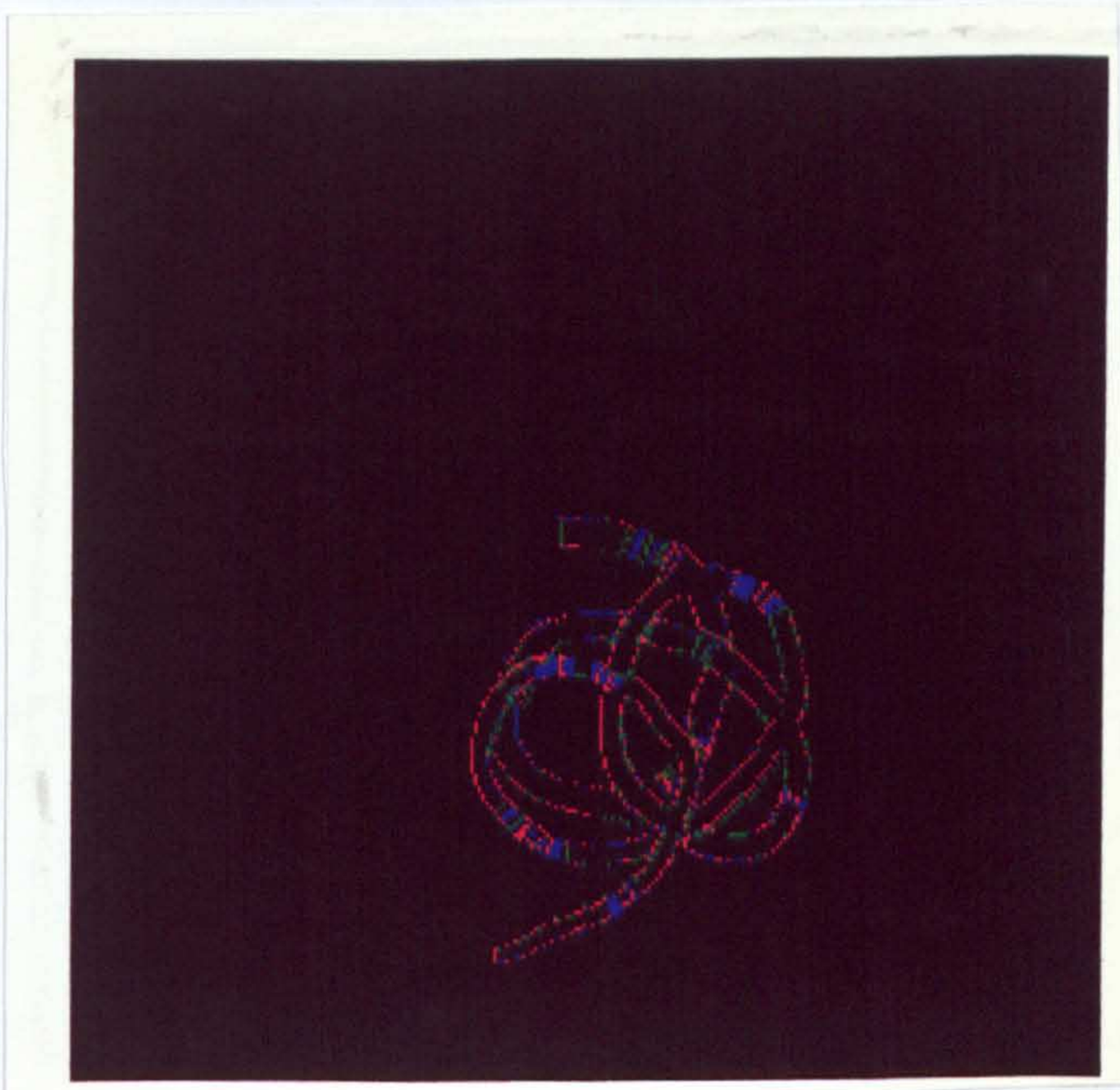

Figure 98: Comparison between learning sets
(extracted from Figure 88 and Figure 90)

Figure 99: Lenna image processed by a 3^2x20x1 network using a learning set extracted from Band A (Figure 88)

5.5.6 Network Size

Several networks with different hidden layer size were tested. and the network with the highest learning rate was selected. If similar learning rates were obtained for a number of networks then the networks were tested on the same image so as to try to select the best network. The smallest network is the preferred one, as it will minimise processing time and the amount of memory required.

There is no known rule as to the best size of hidden layer, or even as to the most suitable first attempt. Some authors suggest the hidden layer have half of the number of input layer nodes. Others suggest a slightly larger number. A trial and error procedure was followed. The first attempt is usually to try the same number of hidden layer neurons as the number of input neurons. Larger and smaller networks are then tried. Usually a larger hidden layer implies easier learning, (although more time for each iteration) but does not imply a better solution or a better learning rate. Indeed there exist an area where the learning achievements of the networks are similar, and for which no evident performance differences are detected in processed images.

Figure 100 and Figure 101 shows the number of learnt patterns for a 3^2 and a 7^2 input network respectively, versus the number of hidden nodes for different numbers of iterations. Unless a very small and clearly unsuitable hidden layer was used then the results obtained were similar. This suggests that there is little dependence upon the performance of the networks with reference to the number of hidden nodes investigated.

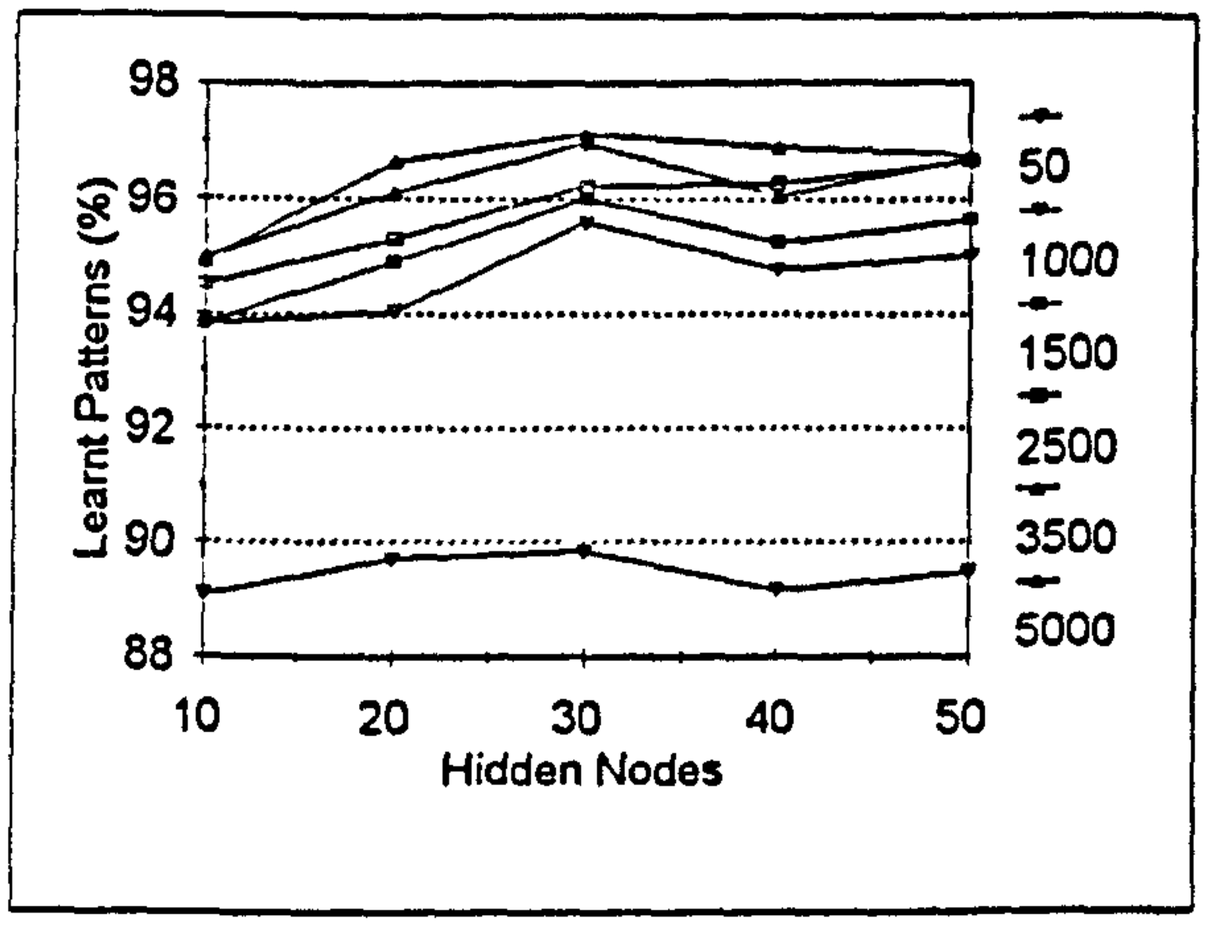

Figure 100 :Evolution of learnt patterns for a 3^2 input network using different numbers of iterations

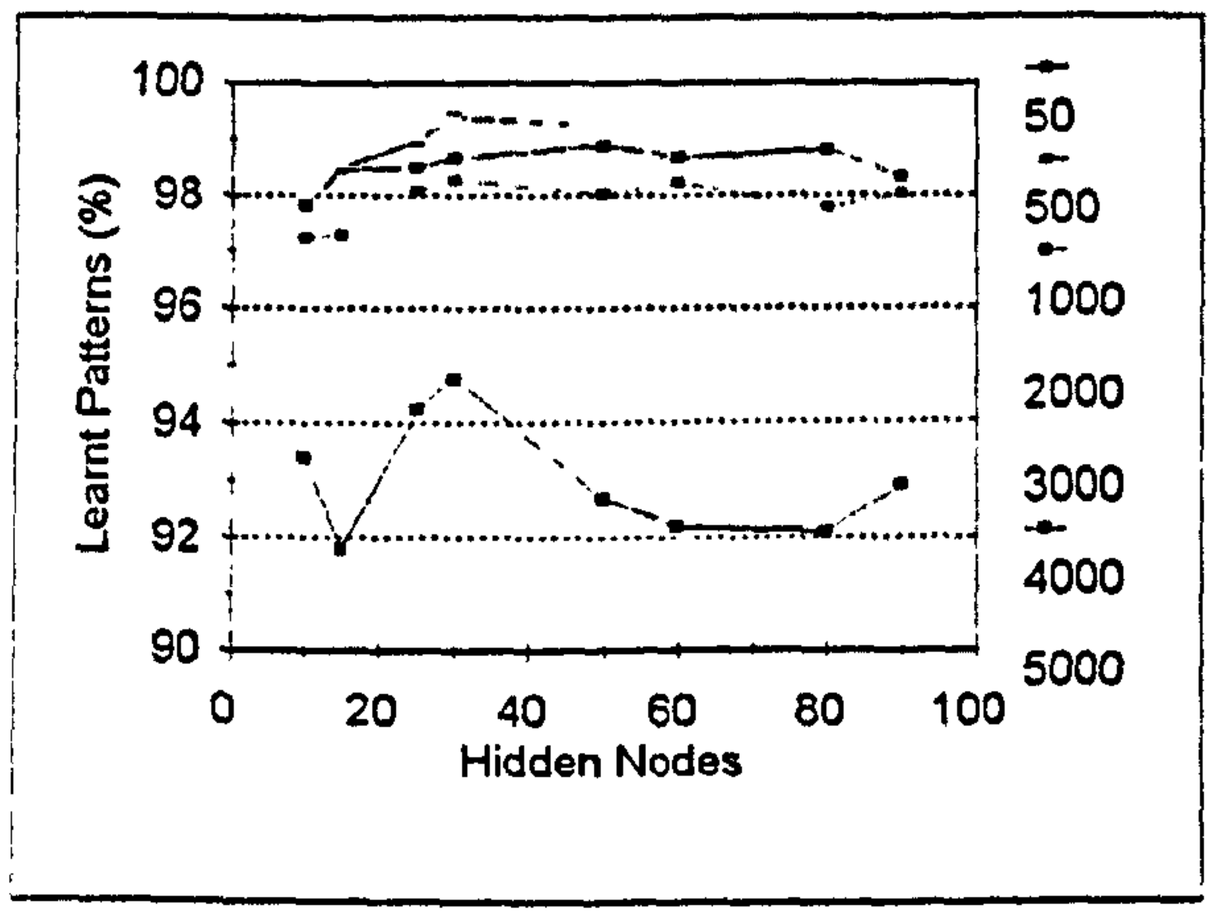

Figure 101 : Evolution of learnt patterns for a 7^2 input network using different numbers of iterations

However. a larger hidden layer always implies a more computationally expensive solution. Figure 102 through 105 show the learning performances of a 7^2 input network having 10. 30. 50 and 70 hidden nodes respectively. These networks were taught using the full learning set extracted from Figure

90 . The networks used are from the ones presented in TABLE XIV - C.

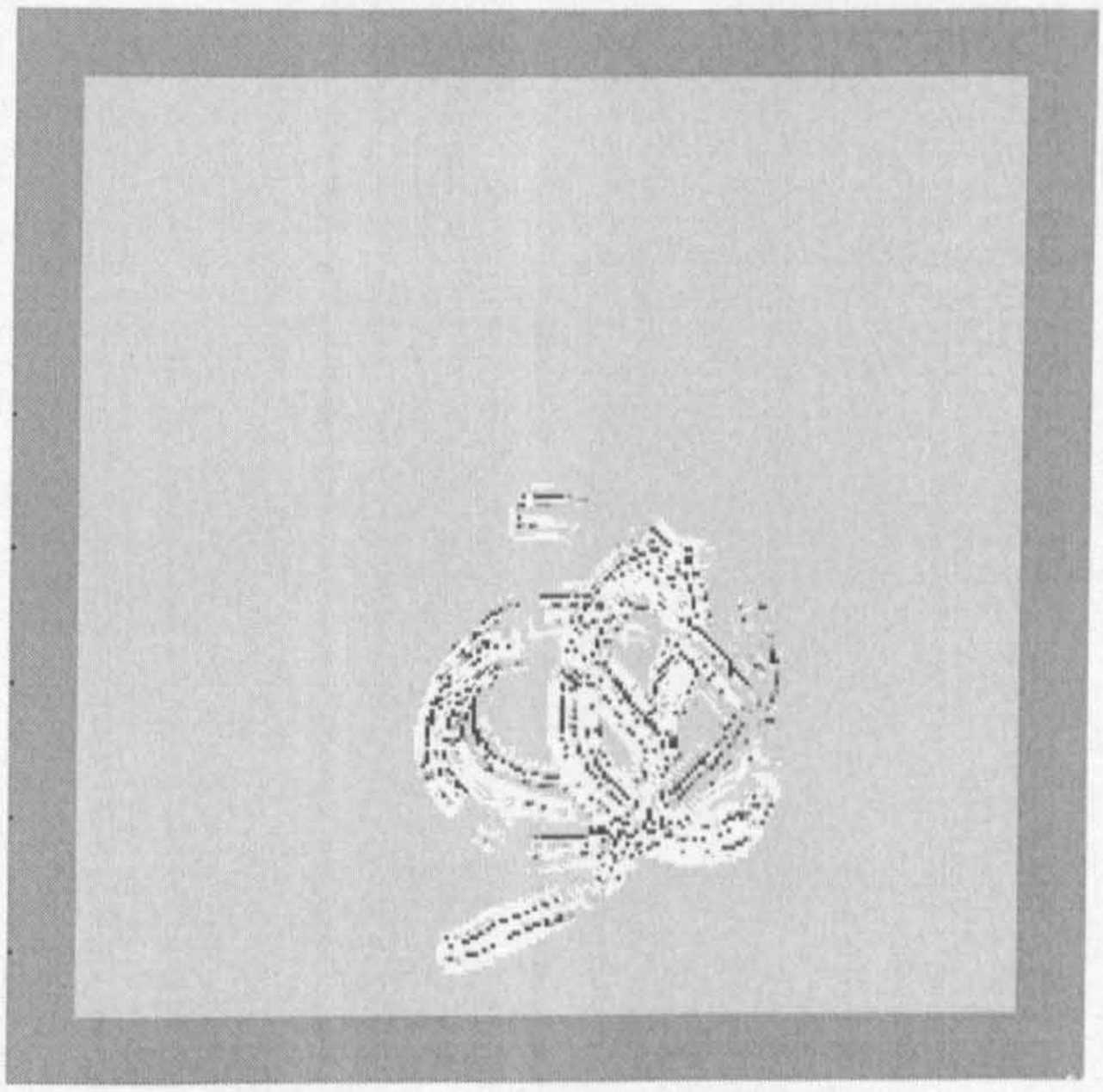

Figure 102: Learning performance for a 7^2x10x1 network

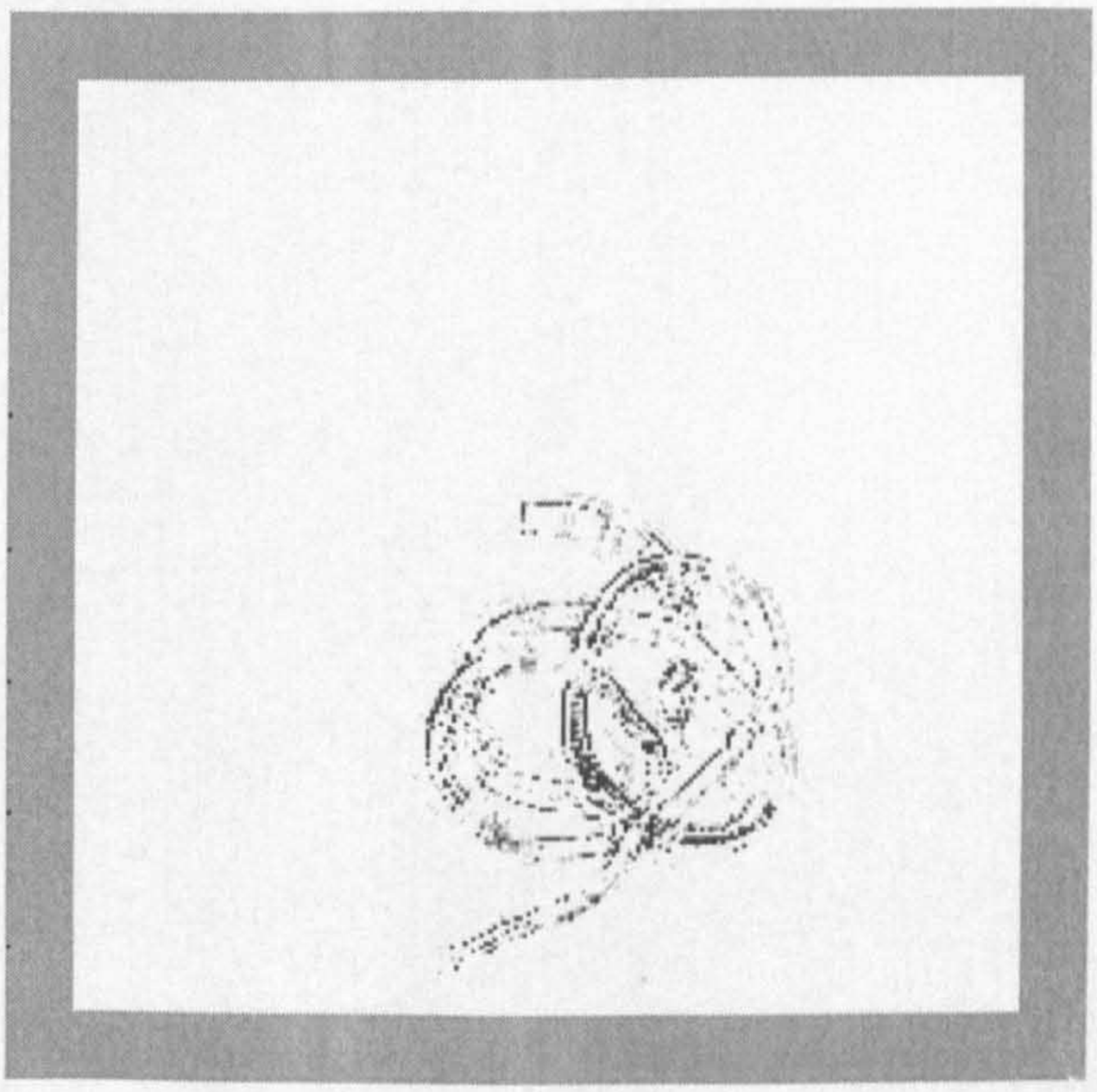

Figure 103: Learning performance for a 7^2x30x1 network

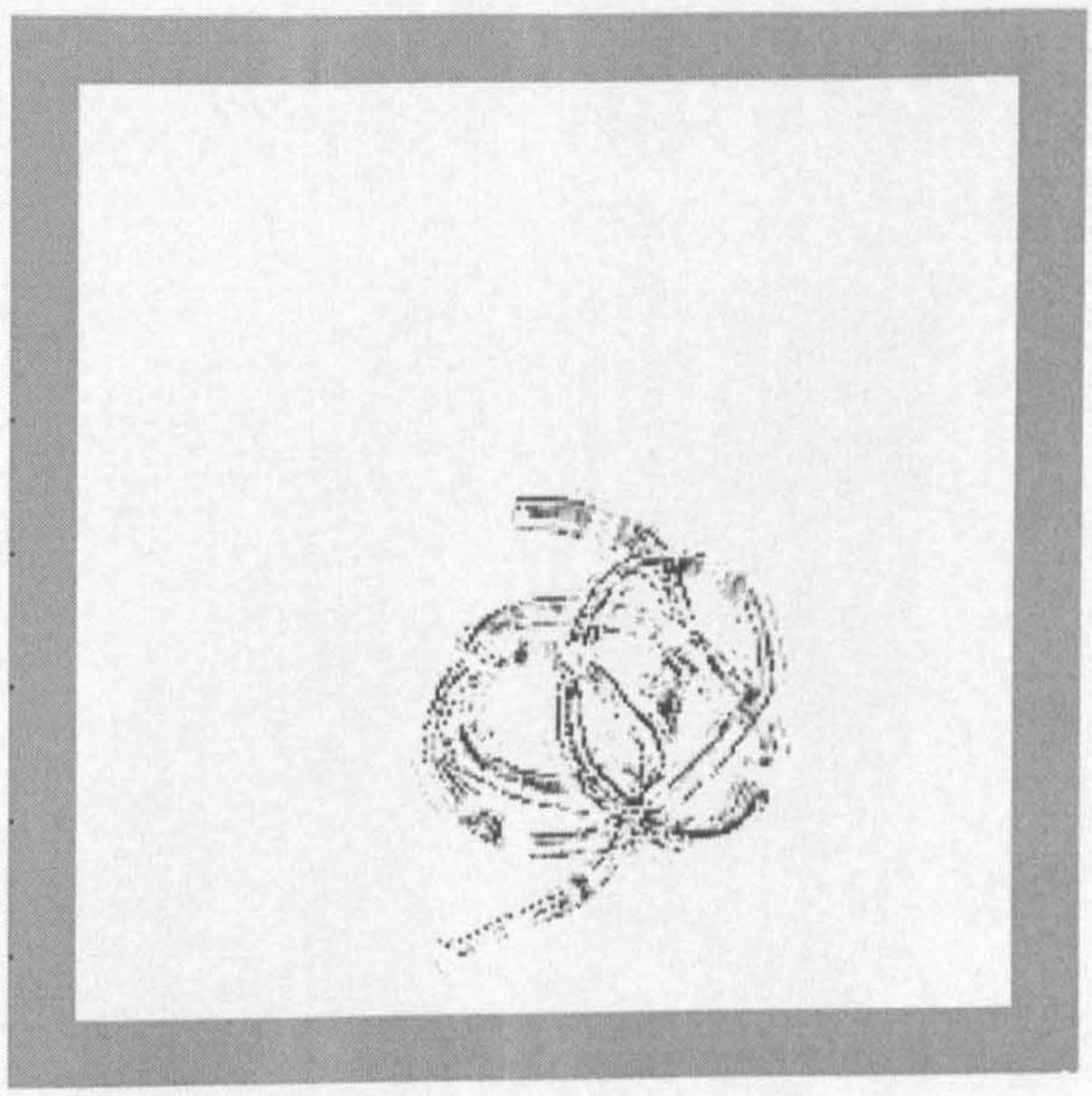

Figure 104: Learning performance for a 7^2x50x1 network

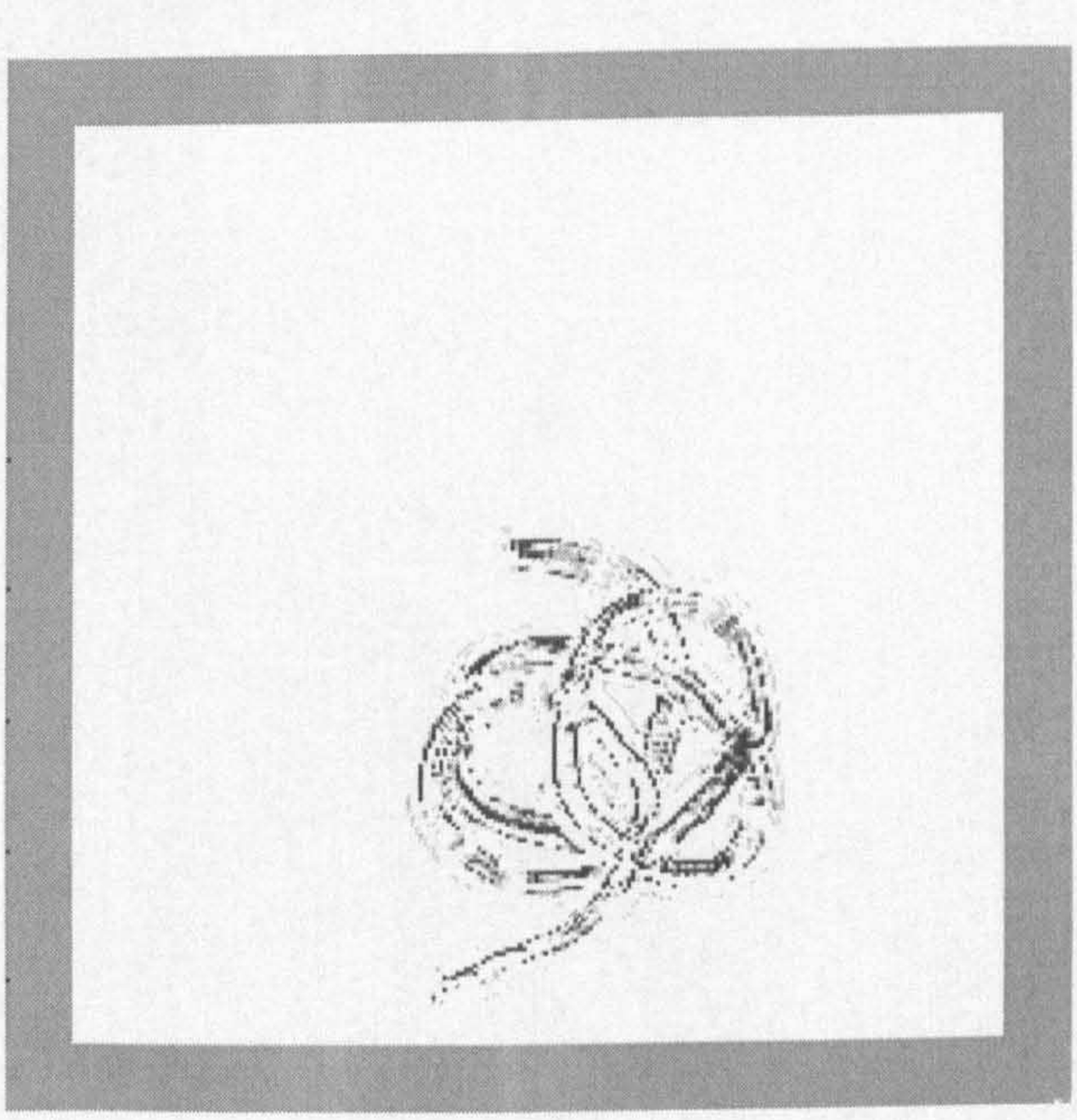

Figure 105: Learning performance for a 7^2x70x1 network

It can be seen that more or less equivalent solution were achieved by all the networks, with only a variation of 3 to 5 % being obtained in the first two

cases and 0,5 to 1,5 % in the second two cases, in the number of unlearnt parameters. In respect to the generalisation capabilities of the network, an example is presented in the following pictures. Two processed girl images, (networks with 3x3 input nodes and 30 (Figure 106) and 40 (Figure 107) hidden nodes) were thresholded and subtracted. The difference image is shown in false colours. Common points are not marked as they include the background points. Points uniquely marked by the bigger network are shown in green and points uniquely marked by the smaller are shown in red. The colours were obtained through simply changing the look up table.

Figure 106:Girl processed by a 3^2x30x1 network
using a full learning set extracted from the Band B image

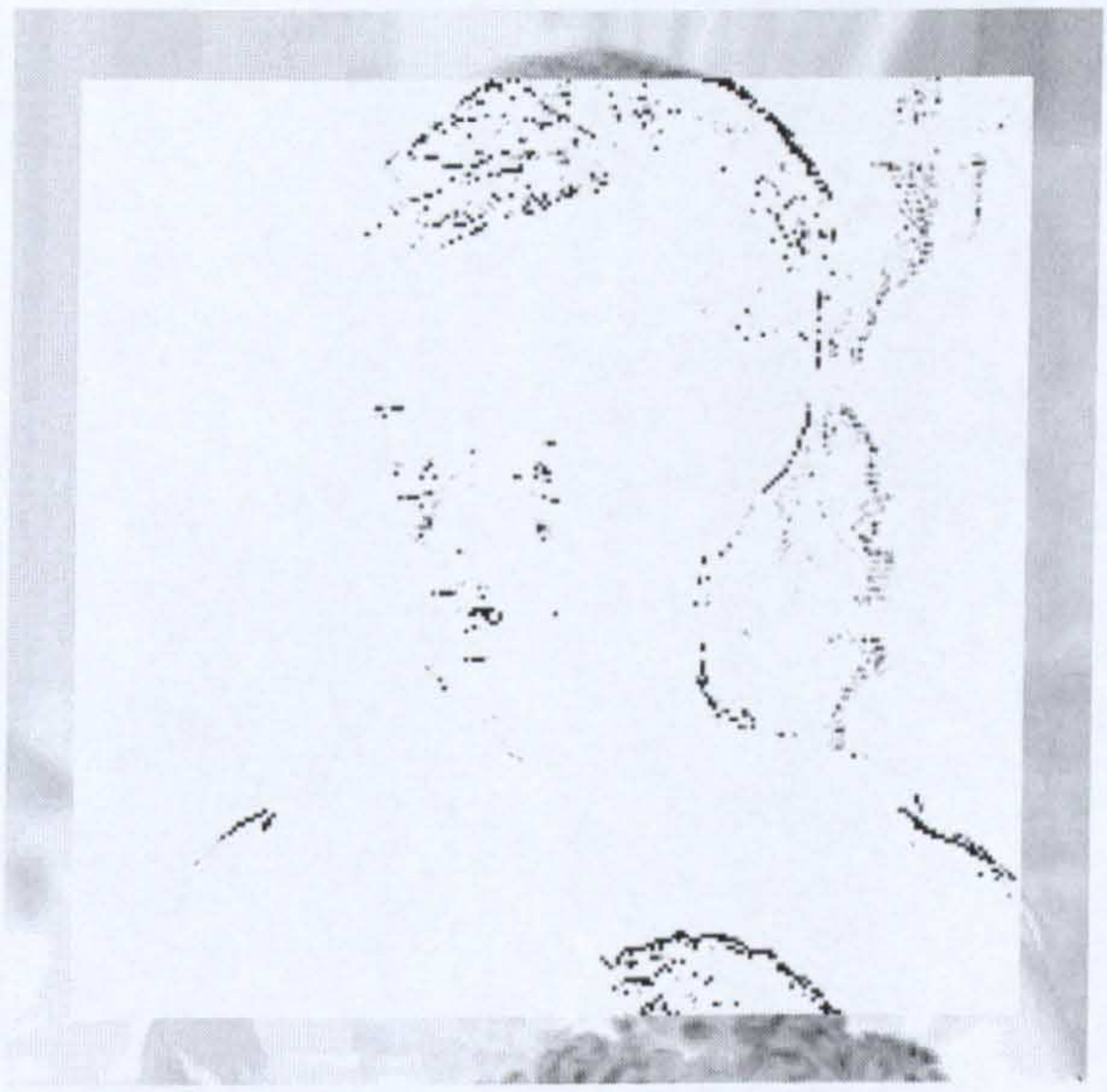

Figure 107: Girl processed by a 3^2x40x1 network using a full learning set extracted from the Band B image

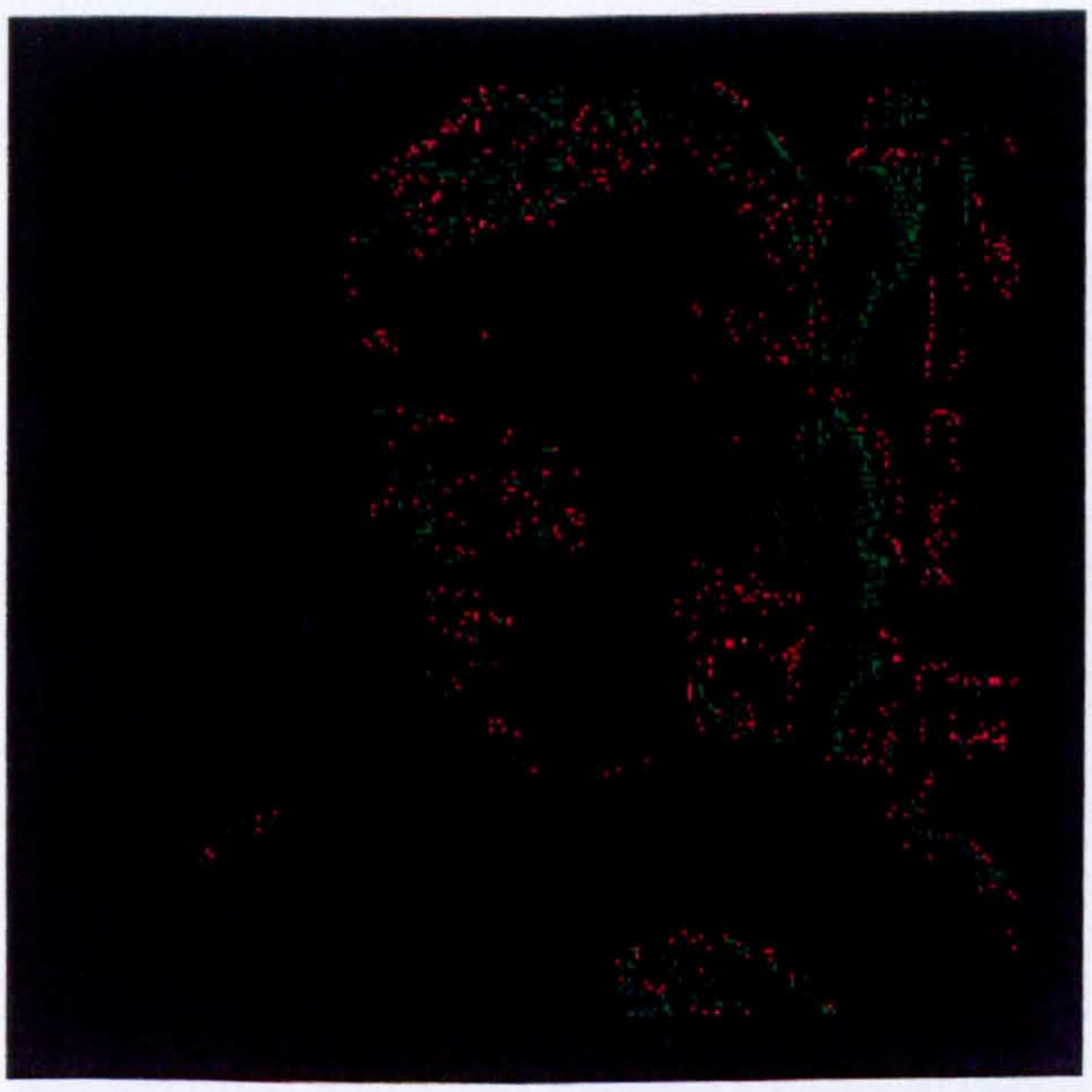

Figure 108: Difference between Figure 106 and Figure 107

Although different, there is no evident advantage of one solution over the other. The only significant difference in performance being observed in the window behind the girl.

5.5.6.1 Full image

A full Band B image was used to extract a full learning set from which other sets of networks were taught. The final values attained are presented in TABLE XXV , and their performance on the bands and squares images presented in TABLE XXVI and TABLE XXVII respectively.

TABLE XXV
Final values obtained with a full learning set
extracted from a full Band B image

Window Size	Final Values			Learning Set Size	
3x3				4,850	
HidNod	WP	RCD	%	@	RMS
10	504	4.346	89.61	3,000	0.09
20	418	4.432	91.38	1,020	0.07
30	402	4.448	91.71	1,340	0.07
40	397	4.453	91.81	1,020	0.07

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXV - A

Window Size	Final Values			Learning Set Size	
5x5				5,761	
HidNod	WP	RCD	%	@	RMS
20	273	5.488	95.26	1,020	0.04
30	263	5.498	95.43	1,020	0.04
40	242	5.519	95.8	1,020	0.04
50	240	5.521	95.83	1,020	0.04

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXV - B

TABLE XXVI
Performance of the neural network edge detectors on the Band Images after training on a full training set of Band B

Window Size.	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3x3	10	0.88	0.57	0.88	0.62
	20	0.84	0.55	0.87	0.66
	30	0.87	0.56	0.9	0.67
	40	0.82	0.59	0.84	0.7

EQ -Edge Quality MQ - Map Quality

TABLE XXVI - A

Window Size.	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5	20	0.82	0.59	0.94	0.78
	30	0.4	0.52	0.95	0.71
	40	0.86	0.55	0.97	0.78
	50	0.88	0.61	0.95	0.81

EQ -Edge Quality MQ - Map Quality

TABLE XXVI - B

TABLE XXVII
Performance of the neural network edge detectors on the Squares Image after training on a full training set of Band B

Window Size.	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3	10	0.7	0.35	0	0.02	0	0
	20	0.85	0.3	0.02	0.07	0.05	0.07
	30	0.81	0.31	0.02	0.06	0	0
	40	0.94	0.34	0.03	0.09	0.06	0.08

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXVII - A

Window Size.	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
5	20	0.77	0.33	0.01	0.02	0	0
	30	0.75	0.31	0.02	0.06	0.01	0.01
	40	0.75	0.34	0.01	0.04	0	0
	50	0.8	0.32	0.01	0.03	0.01	0.01

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXVII - B

5.5.7 Performance

5.5.7.1 Window Dependence

Several windows sizes were tested. Results for three sizes of window are presented in figure 106. figure 109 and figure 110. It was expected that a network whose inputs are from a larger window would perform better than a network whose inputs are from a smaller window. due to the larger area covered in the image which provides more information to the network. It was found that this was not the case. In some cases. the resulting networks marked

erroneously areas with low varying or constant grey level as edges.

Figure 109:Girl image processed by a 7^2x30x1 network
(full learning set)

Figure 110: Girl image processed by a 5^2x30x1 network
(full learning set)

5.5.8 Localisation

It is difficult to characterise the localisation of the edges marked, since some of the networks mark points that are clearly outside the edges. A localisation criteria will always leave in doubt if these points are considered as edges, and as to which of the true (or expected) edges to assign them The points are marked with the same intensity, thus it is difficult to see how strong the categorisation is. When considering localisation, lines considered as singular points should be rejected. Lines, or clusters of points that are perceived as lines, are the most important edges marked. An example is presented in Figure 111, where the edge image was added to the original picture. This was darkened to enhance the marked edge position.

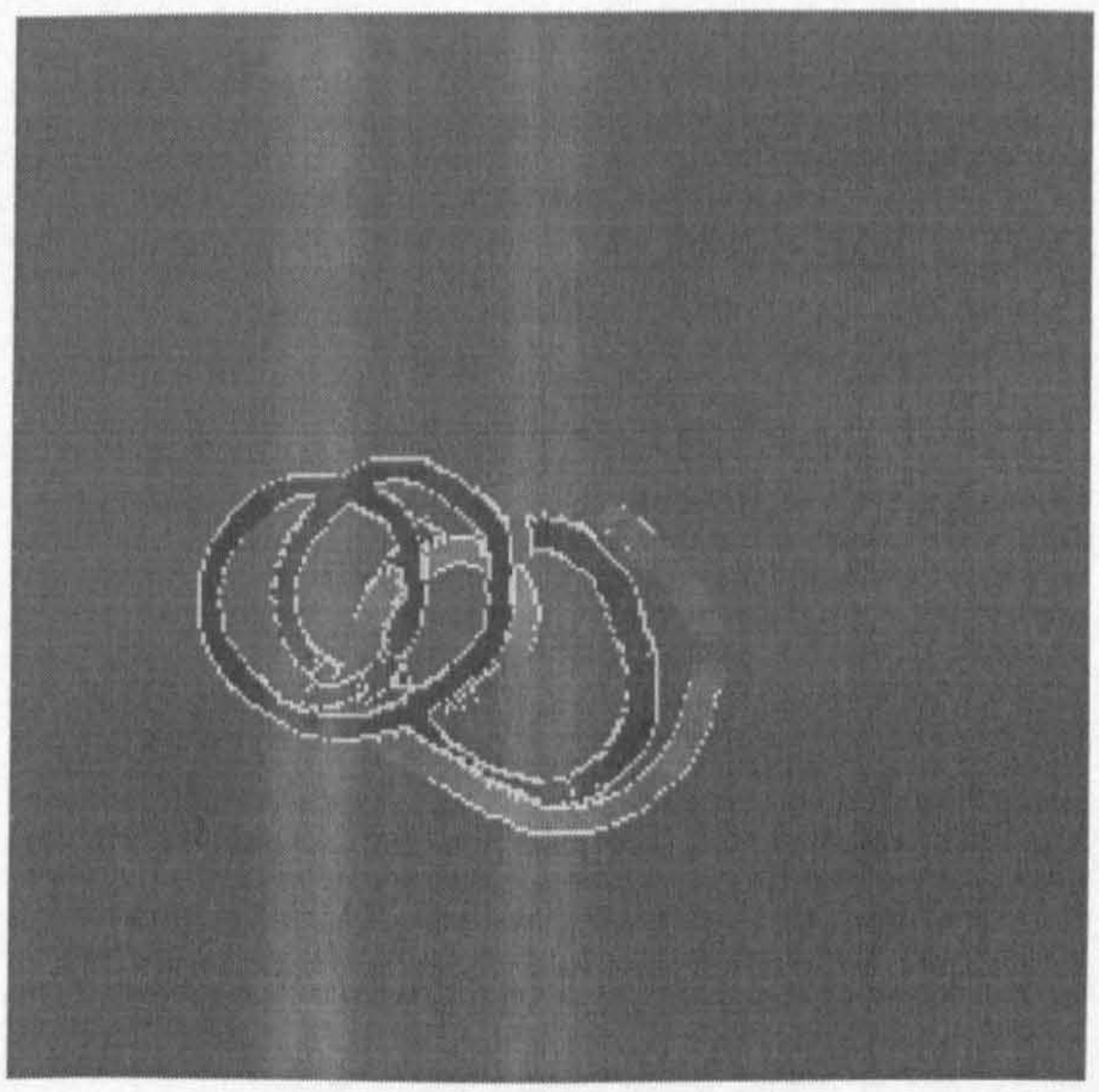

Figure 111: Result of the processing by a 3^2x10x1 network superimposed on the original

5.5.9 Robustness

One of the major advantages of neural networks are their robustness. Indeed it could alone justify the use of a neural solution even if the computational load is considerably heavier. This characterisation is incomplete without verifying how they perform in adverse conditions. To test it a degraded version of the image of Band A was generated. The picture was blurred by a Gaussian function and noise was added. The degraded image is presented in Figure 112.

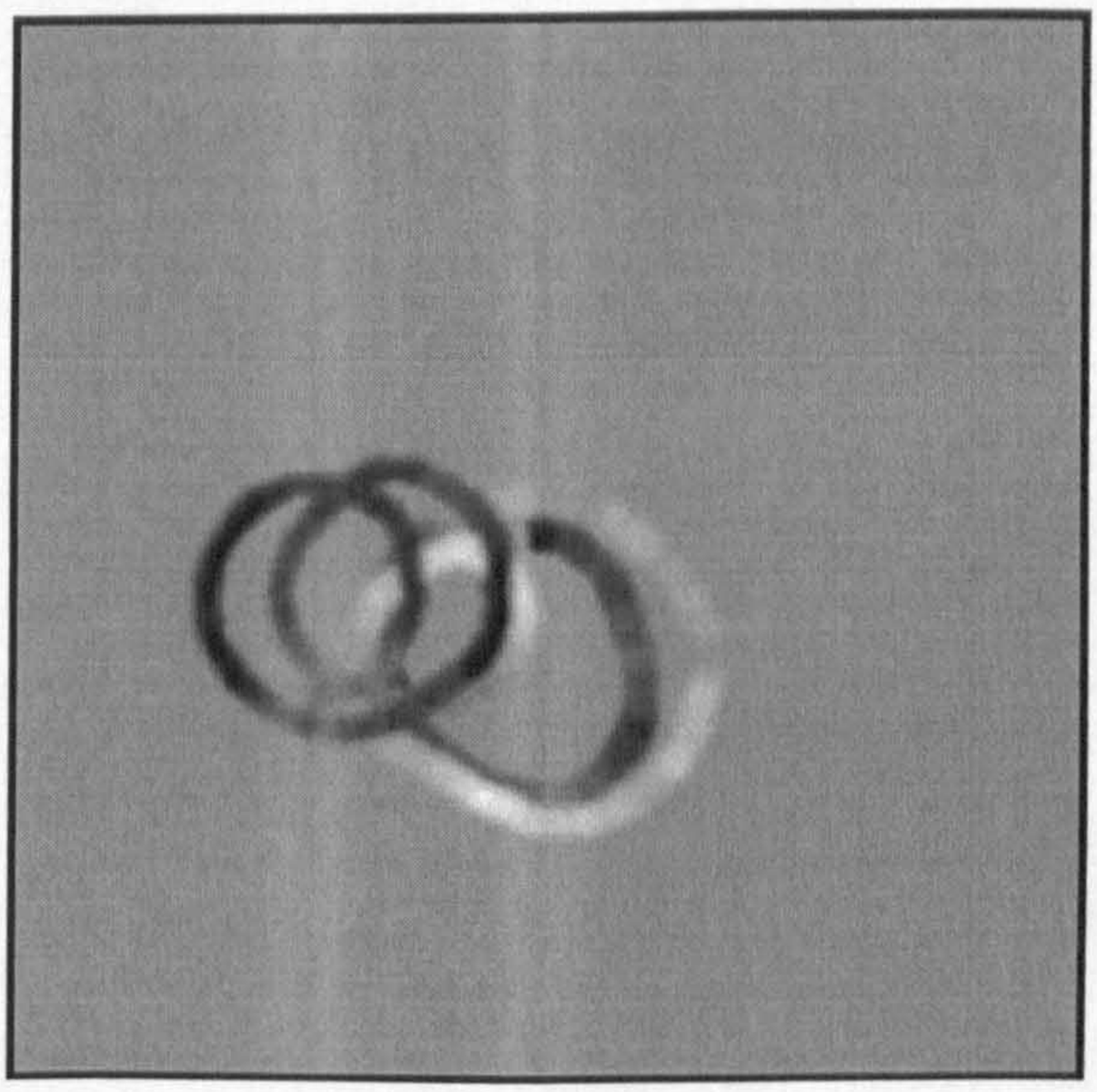

Figure 112: Degraded version of Figure 88

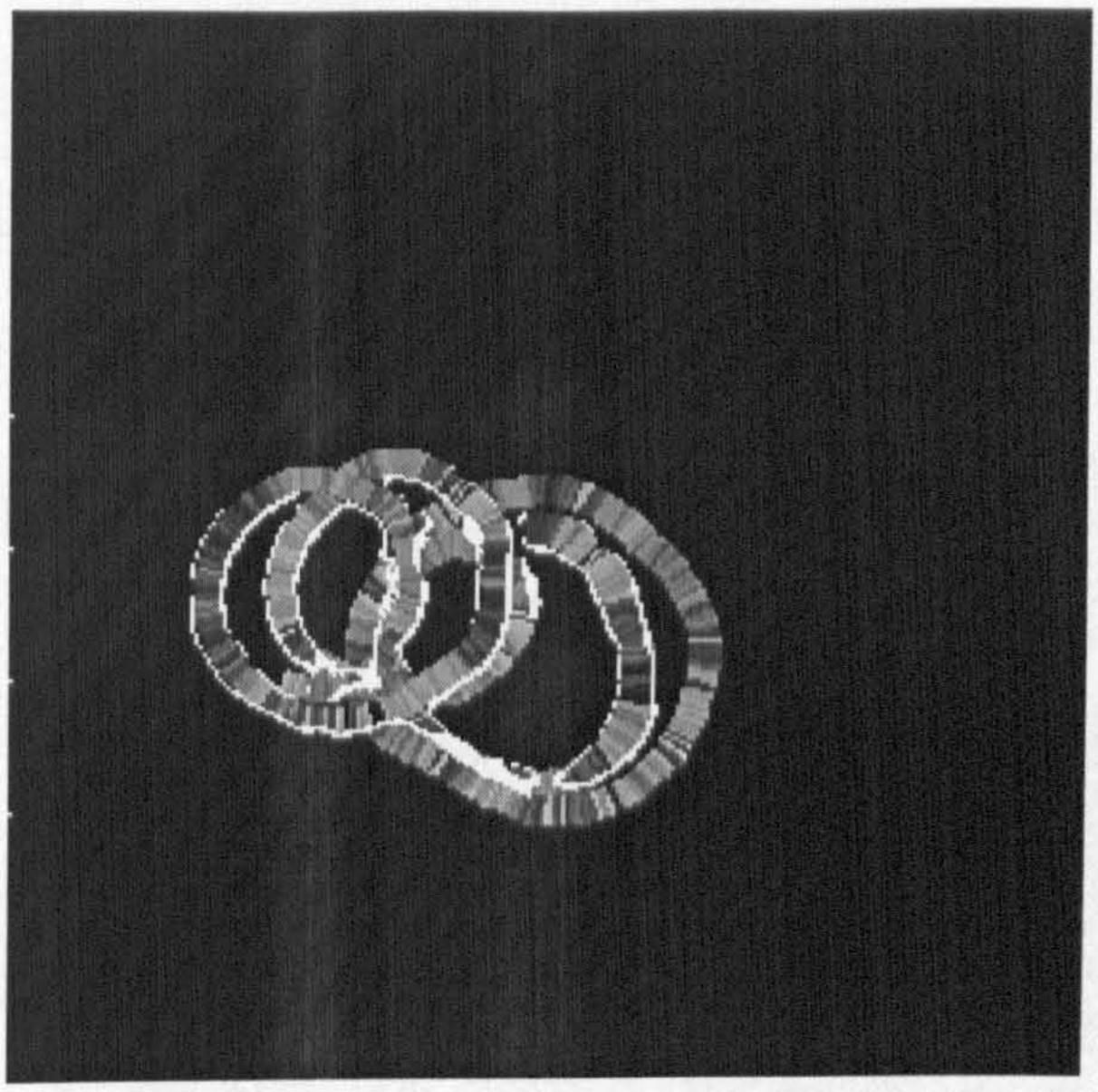

Figure 113:Result of the processing of a degraded picture (Figure 112) by a 3^2x10x1 network superimposed on the original

The image in Figure 112 was then processed with a network consisting of 3^2-15-1 neurons. Figure 113 shows the superposition of the result onto the original picture.

Another way of evaluating the robustness of the solutions is achieved by the definition of a suitable gauge, which could be an image with a linearly variant amount of noise. An example of an image that fulfils this objective is the squares image (Figure 33).

Figure 114 and Figure 115 shows the results of the 3^2x10x1 and 3^2x15x1 networks applied to Figure 33, respectively.

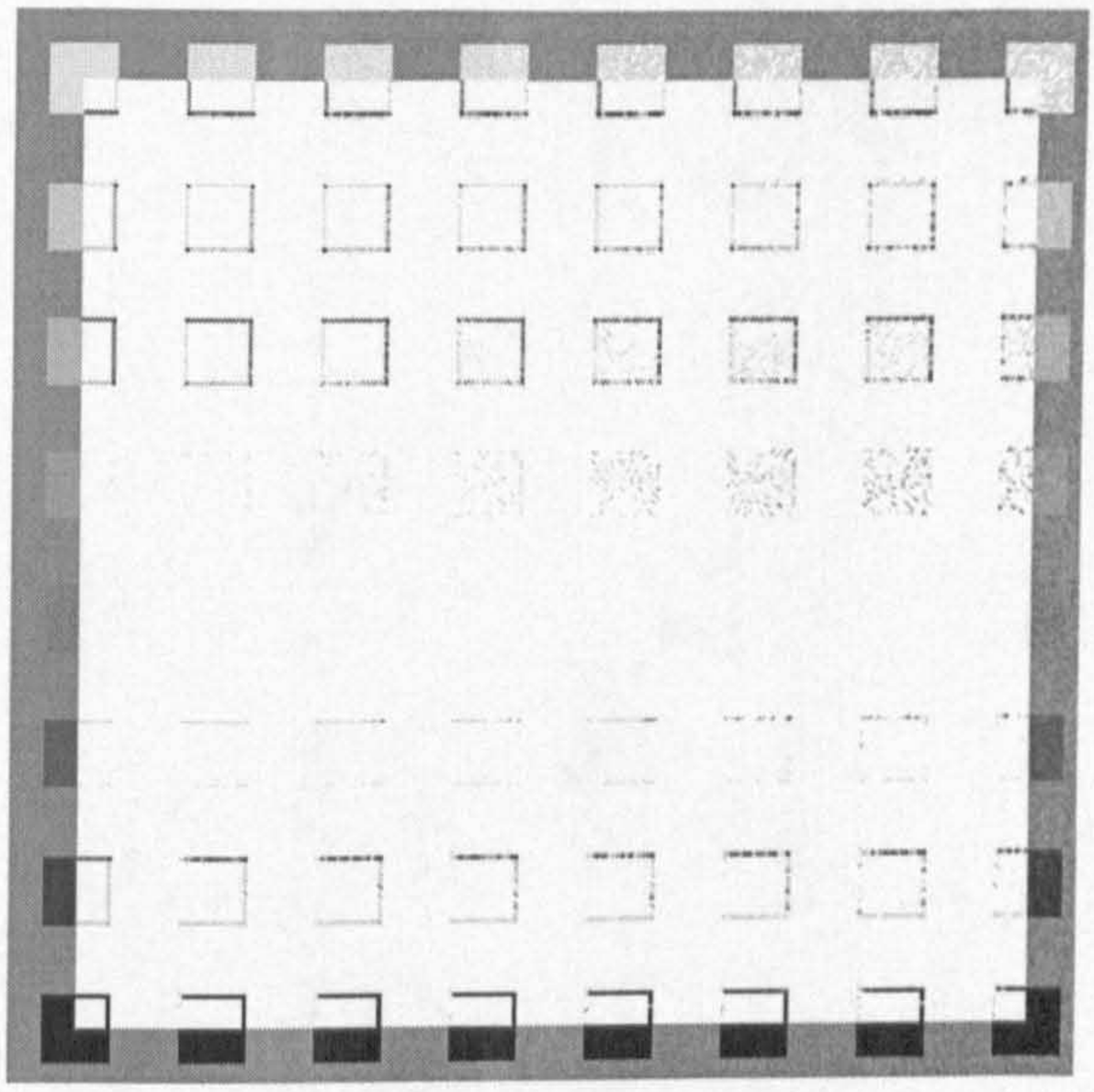

Figure 114: Squares image processed by a 3^2x10x1 network using a reduced learning set extracted from Band A

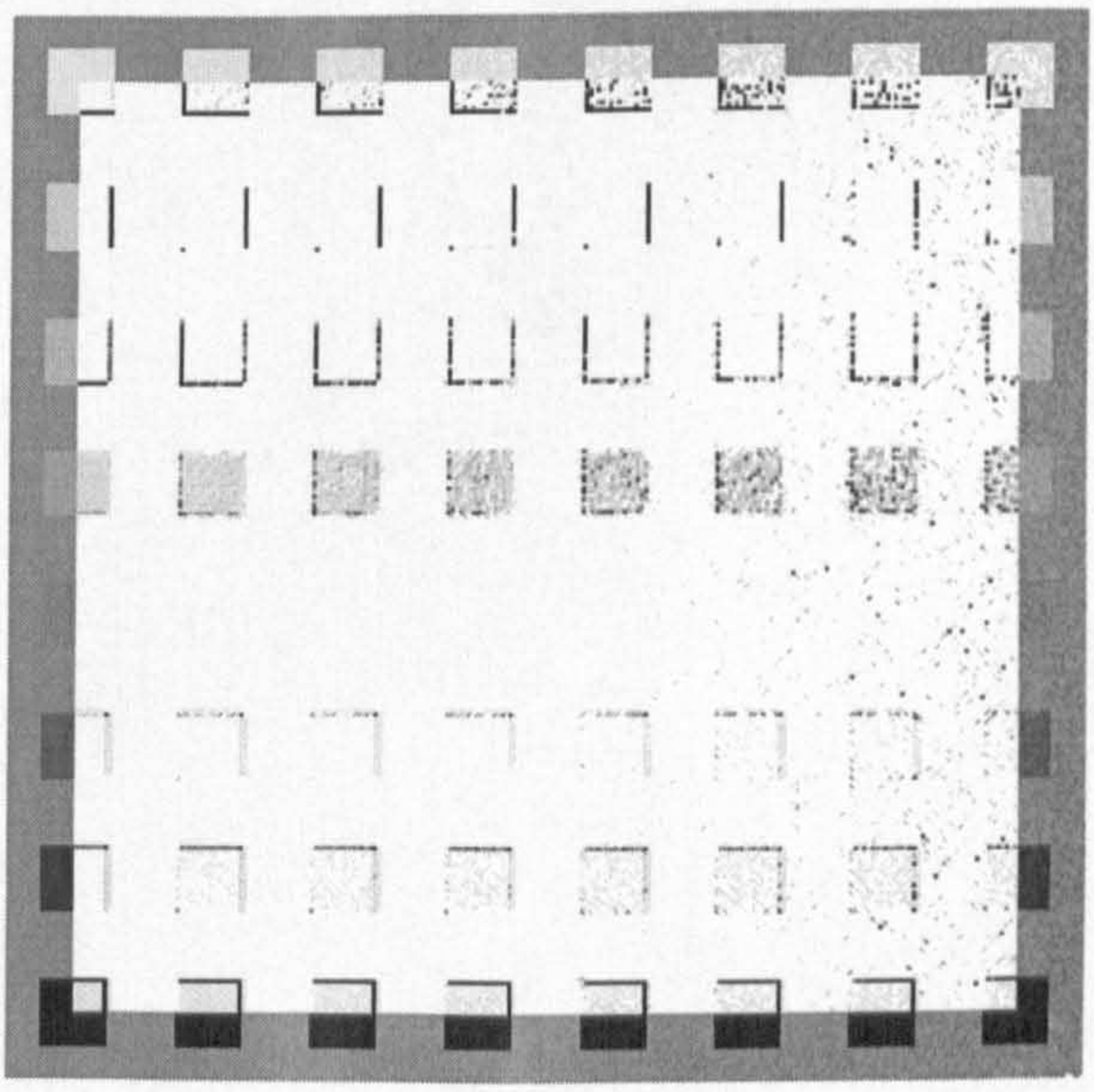

Figure 115:Squares image processed by a 3^2x15x1 network using a reduced learning set extracted from Band A

From these images several interesting points can be noted. Firstly, the

confirmation of the conclusion about the robustness of the networks. Secondly the unexpected factor that the solutions proposed are directional, as the edges are not equally detected in all directions. Also the fact that less contrasted edges are missed.

5.5.10 Computational Load

An important issue in the implementation of image processing algorithms is the computational load. Images require to be of high resolution to be of any use in many image processing applications. The number of operations performed by a neural network are greater than the number of operations which are required by the majority of the algorithms described in the first part of this chapter. However, neural networks include a decision criteria, as to the marking of the points, which will alleviate the computational load of the resulting solutions. This was not active when producing some of the images, so as to maintain the same conditions for the visual comparisons performed. Effectively it is easy to compute a thresholded output for the neural network.

Secondly convolution operators can be implemented in a neural network form. A 3x3 mask corresponds to a 9x2x1 topology with an unitary slope line as the activation function. The networks investigated are inherently non-linear and thus have more potential than a convolution mask.

5.5.11 Discussion

In this section an approach for a neural network edge detection scheme has been presented. The performance of the more effective solutions obtained were

presented together with the options and assumptions made. These form guidelines for the arbitration strategy developed in the next chapter.

Some of the limitations of the approach used were highlighted and are common to the next chapter. There are several variations that can improve the performance of the system. One could be to include an iterative procedure for misclassified patterns to be included in a progressively extended learning set. Some pattern clustering scheme should also be researched, mainly over the targeted images, in an attempt to provide smaller learning sets.

5.6 Conclusion

This chapter has presented examples of edge detection schemes, to which an arbitration strategy may be applied and expected to perform satisfactorily. The second part of the chapter described investigations carried out into edge detection using neural networks. The performance of the solutions achieved were presented, through representative examples chosen from among the converged networks, and the results were discussed. These will be used as a comparison to the arbitration strategy to be presented in the next chapter. The solutions obtained in the next chapter will be presented and their performance characterised. Finally, comparisons between the proposed approach and the presented approaches will be drawn.

6 ARBITRATION

"Pluralitas non est ponenda sin necessitate" ("Occam razor")[14]

6.1 Introduction

In the last chapter some common edge detectors were described and their behaviour and performance analysed in the form of processed images. The development of neural networks for edge detection was presented as was the performance achieved by these networks. Advantages and disadvantages of the proposed solutions were highlighted relative to the previously implemented methods.

The aim of the research is to investigate image edge map improvement by the use of arbitration between different edge detection maps. In this chapter the development of the arbitration process is described, and options and assumptions assumed justified. Examples of arbitration processes are presented. Finally comparisons will be performed and conclusions drawn.

[14] "No more things should be presumed to exist than are absolutely necessary"

6.2 The arbitration system

The general form of the arbitration system is represented in a block diagram form in Figure 116

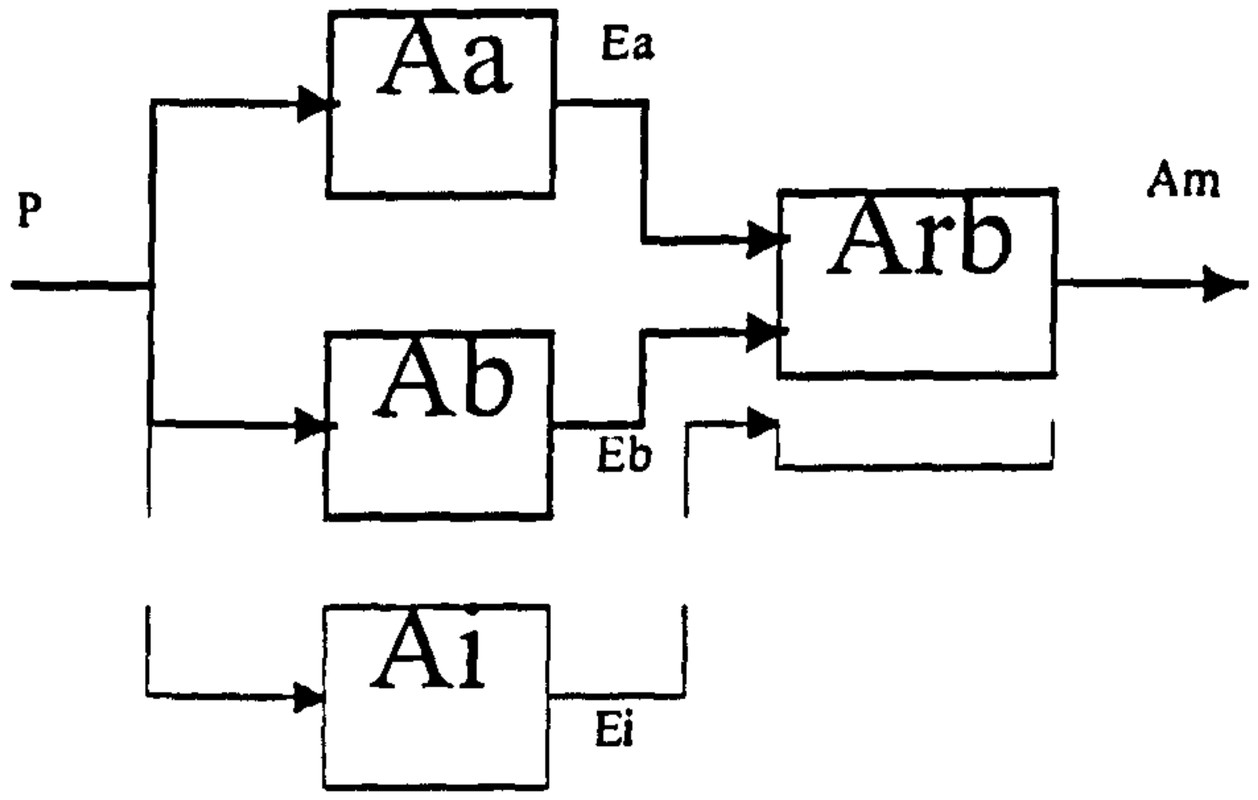

Figure 116 : Arbitration Strategy

Firstly the image P is processed by a set of edge enhancement filters Aa. Ab ...Ai. The output of each of these edge enhancement filters are edge enhanced maps Ea. Eb, ... Ei. From these maps, edges are arbitrated by a Neural Network. based on the information provided by the input data Ea ... Ei. to produce an edge map Am.

A slightly different version could be designed. if we substitute the edge enhancement filters by edge detection filters. The arbitration system will process. in this case. binary vectors instead of multi valued vectors. These will provide less information for the arbitration system. than the one present in our system. as information regarding edge strength is lost. Edge strength is one indicator of the importance of an edge. as in general cases main objects within

a scene are the ones which present high contrast or are represented by high SNR components of the image. However, when assigning patterns to the decision classes of edge / non-edge, mapping inconsistencies are generated. This suggests that more information than that provided by thresholded edge maps is necessary. An example can be taken from the arbitration between two edge maps with marked lines wider than the window size used for arbitration. In this case in the area covered by such lines, the same pattern (all pixels have the value 1) must be assigned to an edge and a non-edge class simultaneously. This case will require larger windows, or additional information to that provided by thresholded edge maps only.

The arbitration process has some similarities with the neural networks for edge detection process described in the last chapter. The core difference is that in the arbitration process. the network processes data extracted from more than one window, in paired positions, from more then one distinct edge enhanced map.

The number of algorithms used in the tests carried out, due to hardware limitations, were kept to two. The hardware requirements for the edge detection case are. at least, doubled for the arbitration case. However, the principles applied can be extended to any number of algorithms.

Effectively, and as explained later in this chapter. an increase in the number of methods to be arbitrated. will produce problems that the hardware available could not handle in an admissible time. The memory limitations will be aggravated and the network learning phase will increase. This is incompatible with the time limitations imposed on the current research.

6.3 Implementation

A general outlook as to the implementation procedure was presented in Chapter 5. The arbitration procedure and the edge detection procedures were developed in parallel as basic options are common to both cases, e.g., learning set source. Emphasis will be placed on main differences and particularities of the neural network arbitration implementation relative to the neural network edge detection implementation, rather than repeat common options assumed during development.

The main difference is the number of hidden nodes. as the networks input size is multiplied by the number of methods arbitrated upon. However, as stated in chapter 5, the number of hidden nodes does not seem to be a decisive factor in the performance achieved. Networks with different hidden node number will be tested. In practice processing times for the arbitration of 1 image changes from the order of tens of minutes to several hours for the larger networks tested. This factor is a main limitation to extensive tests being performed on very large networks.

The time necessary to process one image is a second limiting factor to the number of methods being arbitrated upon. In addition to the problems that exist with the learning phase. processing will be extremely slow as a larger transputer network is not available. Although the processing time could be reduced. as some edge information is already present when the arbitration is started. This will reduce the time used to process the band images. however some thresholding would be necessary for the other images used throughout the work. The processing could be reduced to the areas where edge enhanced maps are non null. This option of the implementation of an empirical shortcut was avoided as it could produce a bias in the comparisons performed. due to

false or incorrect edges hiding in the non processed areas.

6.3.1 Image Selection

The images used throughout the development in Chapter Five were the ones used in the work reported in this chapter. This simplifies the development phase as it allows direct comparisons to be performed. The images used are presented in Figure 32. Figure 34 and Figure 90. The neural network selection options are common with the ones presented in Chapter Five .

6.3.2 Learning Set Extraction

The learning set extraction follows a similar procedure as that used for the edge detection networks. The selected area to be scanned is the widest meaningful area from the edge maps involved. This process is illustrated in Figure 117. First the image is scanned to define the region of interest from which the learning vectors will be extracted. This area is defined as in Chapter Five. All the edge enhanced maps are scanned and the widest area chosen. Around this area a border is added. in order to allow windows to work. This area is then scanned. using a previously defined step, for the pattern extraction. From each of the edge maps, the pixels inside the window are extracted and concatenated to form a pattern. This is stored together with the information as to whether it represents or does not represent an edge. A set of vectors is then produced and stored. which will be presented to the neural network for the learning process. Vectors including only background pixels are discarded. with the exception of the first. following the same procedure as that used for neural network edge detection.

Learning sets are larger than those generated for the cases reported upon in the previous chapter, as the area covered by the edge maps is wider than the bands themselves.

6.3.3 Topology

For the arbitration strategy the number of network inputs is doubled when compared to the neural network for edge detection. Taking as a starting rule that the number of hidden nodes should be similar to the number of input nodes, the memory needs to quadruple relative to the previous case. This factor made using the transputer system difficult. This prompted the migration of the learning program to the Vax/vms system. Some of the presented networks were taught in multiple batch jobs, successively submitted to the lower priority batch queues, as each iteration required more CPU time than that allowed for medium priority ones or running on-line.

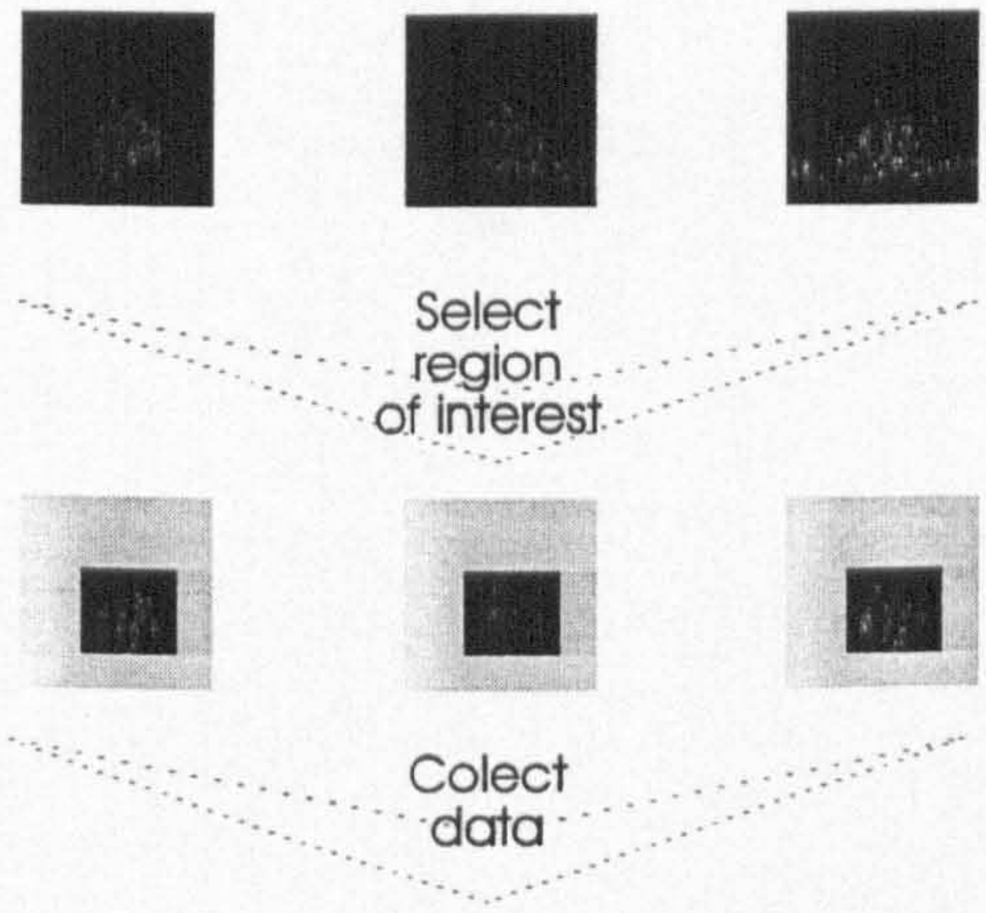

Figure 117:Learning set definition

6.4 Procedure

Images were used as presented, without any further processing or thresholding. Despite the lookup table used for displaying and or printing, edges are marked proportional to their strength. This affects some of the manipulations to be carried out.

For the arbitration cases a hard threshold was introduced into the network in order to produce binary pictures as the output.

6.5 Performance

6.5.1 Generalities

The following sections describe tests performed using different arbitration schemes. Firstly the most relevant characteristics of the methods are described.

The networks used are presented along with their learning achievements and the characteristics of the learning sets used. These are presented together with at least one of the images processed. The complexity of the images does not allow quantitative measures, that fully reflect the edge map quality, to be evaluated. Thus the same indicators will be used as before where generated images are used for both learning and in the testing of the various arbitration schemes. This makes it difficult to distinguish which network performed the best. However, in some cases, failed edges are different. The processed images chosen to be presented are the ones which seemed to be the more representative solution from a visual aspect, providing more defined edges and a minimal number of scattered points or misplaced lines. The measures

performed in chapter five are repeated for the images presented, and these values will be considered for quantitative comparisons.

The "Girl" image was again the best image to reveal weaknesses and differences, and thus is presented in the majority of cases.

6.5.2 Case A - Roberts vs. Canny

6.5.2.1 Methods

The first test of the arbitration strategy will be performed between the Roberts (see section A4) and Canny (see section A11) edge detection operators. Roberts, as the quickest, allows the definition of the most sharp edges. However it fails to define low contrast ones. As the smallest window used, and thus with the smallest embedded smoothing, it also has a quite remarkable sensitivity to noise. Also as a gradient operator, it will mark a wider edge in proportion to the slope between the two adjacent areas with a near constant grey level. The Canny operator marks edges as a line 1 pixel wide, but does not distinguishing between their relative strength. This information can however be extracted from the slope at the zero crossing of the second derivative and is present in the edge enhanced map used. The Canny operator can originate a bias in edge localisation which depends on embedded smoothing. A small amount of this is present in the image used. It is expected that the bias will be corrected. Also a systematic misdefinition of the corners in the squares images are expected to be corrected.

These two operators present opposite characteristics. The Roberts operator will only mark a few strong edges whilst the Canny operator will mark both weak and strong edges. Edge maps used for generation of the learning sets are

presented in Figure 118 and Figure 119.

Figure 118:Band B processed by the Roberts operator

Figure 119:Band B processed by the Canny operator

These are used together with the reference map presented in Figure 91 to generate the learning set.

An image of the differences between the two edge maps, highlighted by false colours, is presented in Figure 120. The Roberts algorithm is unique from amongst the ones investigated as it is the only one not centred over a single pixel. Thus it leaves an arbitrary choice as to the location to mark as an edge. This results in with an error of $\pm\frac{1}{2}$ pixel in either direction. These originate the blue border in the false colours picture.

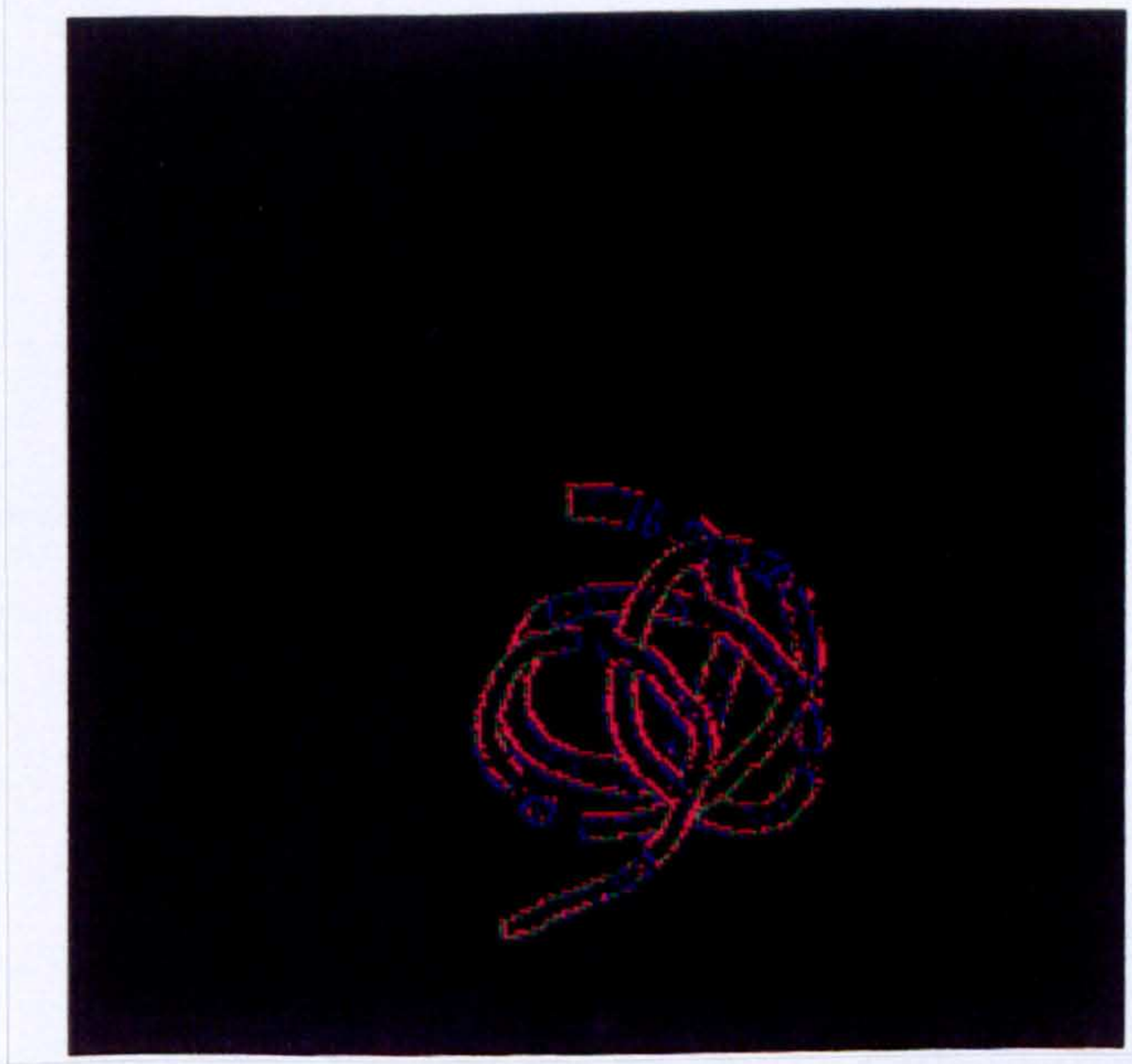

Figure 120:Difference, using false colours, between Figure 118 and Figure 119

TABLE XXVIII presents the size values of the learning sets used for differnt window sizes.

ROB+CAN vs. REF	region of interest 84 x 110	
window size	learning sets	
	used	region of interest
3x3	5,546	9240
5x5	6,283	
7x7	6,807	

TABLE XXVIII
Learning Sets characteristics
for Roberts vs. Canny Arbitration

Vectors containing only the background are discarded which accounts for 26% to 40% of the region of interest.

6.5.2.2 Performance when applied to the Bands and Squares images

TABLE XXIX presents the learning rates for the various networks for different sizes. TABLE XXX and TABLE XXXI present the measures over the processed images. The best solutions are obtained for the smallest networks. as can be concluded from comparison between the images processed by the different networks. The smallest windows do not present a smaller number of scattered isolated points as they are able to mark main edges with smaller distortions.

TABLE XXIX
CASE A
Final Values cbtained for the neural network arbitration
between the Roberts and Canny edge maps

Window Size	Final Values					Detail
3x3						
Hidden Nodes	W P	RCD	%	@	RMS	E
5	866	4,680	84.39	650	0.01	21
10	642	4,904	88.42	500	0.94	15*
15	498	5,048	91.02	600	0.08	15
18	504	5,042	90.91	450	0.77	16
30	427	5,119	92.3	450	0.69	16
40	268	5,278	95.17	600	0.05	14
50	296	5,250	94.66	500	0.52	25
WP - Number of wrongly recalled patterns			@ - Iterations for convergence			
RCD - Number of correctly recalled patterns			RMS - Root mean square error at convergence			

* Better defined shape

TABLE XXIX - A

Window Size	Final Values					Detail
5x5						
Hidden Nodes	WP	RCD	%	@	RMS	E
25	133	6,150	97.88	750	0.02	23
50	100	6,183	98.41	850	0.02	20
WP - Number of wrongly recalled patterns			@ - Iterations for convergence			
RCD - Number of correctly recalled patterns			RMS - Root mean square error at convergence			

TABLE XXIX - B

Window Size	Final Values					Detail
7x7						
Hidden Nodes	W.P.	RCD	%	@	RMS	E
20	85	6,722	98.75	840	0.01	20
50	53	6,754	99.22	280	0.01	25
WP - Number of wrongly recalled patterns			@ - Iterations for convergence			
RCD - Number of correctly recalled patterns			RMS - Root mean square error at convergence			

TABLE XXIX - C

TABLE XXX
CASE A
Performance of the arbitration process between the Roberts and Canny edge maps when applied to the Band Images

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3	5	0.62	0.35	0.63	0.42
	10	0.64	0.35	0.63	0.4
	15	0.64	0.35	0.64	0.43
	18	0.64	0.36	0.64	0.42
	30	0.56	0.35	0.57	0.41
	40	0.6	0.3	0.57	0.35
	50	0.57	0.33	0.56	0.4

EQ -Edge Quality MQ - Map Quality

TABLE XXX - A

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5	25	0.45	0.27	0.74	0.7
	50	0.61	0.36	0.98	0.91

EQ -Edge Quality MQ - Map Quality

TABLE XXX - B

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
7	20	0.53	0.34	0.99	0.93
	50	0.47	0.3	0.98	0.92

EQ -Edge Quality MQ - Map Quality

TABLE XXX - C

TABLE XXXI
CASE A
Performance of the arbitration process between the Roberts and Canny edge maps when applied to the Squares image

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3	5	1.12	0.63	0.01	0.02	0.03	0.03
	15	0.87	0.44	0.01	0.02	0.03	0.03
	18	0.98	0.5	0.01	0.01	0.03	0.02
	30	1.11	0.5	0.02	0.02	0.05	0.03
	40	0.82	0.32	0.02	0.02	0.05	0.04
	50	0.94	0.42	0.02	0.02	0.05	0.04

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXXI - A

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
5	25	1.05	0.39	0.06	0.04	0.12	0.07
	50	0.95	0.34	0.05	0.04	0.1	0.06

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXXI - B

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
7	20	0.91	0.21	0.07	0.04	0.13	0.04
	50	0.99	0.25	0.04	0.03	0.08	0.04

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXXI - C

6.5.2.3 Position

Figure 121 and Figure 122. show superimposed on the original girl image the edges marked by networks with 3^2 and 7^2 input nodes respectively. As in the neural detection case. edges are marked incompletely and in the correct position .

Figure 121:Arbitration using a 3^2x5x1 network superimposed on the girl image

Figure 122:Arbitration using a 7^2x5x1 network superimposed on the Girl image

6.5.2.4 General Performance

A final comparison should be performed between the obtained edges from the arbitration system and the edges obtained by the edges operators alone. It is important to assess what and where are the gains obtained using the arbitration strategy. The images in figure 123 and 124, and the images in figure 125 and 126 present the results from the blending of the arbitrated edge maps and the edge maps arbitrated upon. In this case lost edges are marked in red and gained edges are marked in green. Edges that are present in both images are marked in blue. The arbitration strategy is performing more than a logical *AND* .If the arbitration strategy only perform a logical AND only common points to both edge maps would be marked, thus the comparisons would only present common and lost points (or blue and red points). This fact can be concluded from the existence of green points, as the only non common points would be the ones that are only present in the original image. The fact that red points exist show that more than a logical *OR* is being performed. As in this case points marked by one or more of the operators would be marked in the final edge map. Thus the blending would originate only common and gained points (or blue and green points).

The 5^2 window produced some of the best recalling. Here it is difficult to see the differences between the processed image. from which the learning set was extracted. and the reference image used in the extraction. These are presented for one of the cases studied. in Figure 127.

Figure 123:Blending of the edges produced by a 3^2x5x1 neural network arbitrator and the Roberts operator results

Figure 124:Blending of the edges produced by a 3^2x5x1 neural network arbitrator and the Canny operators results

Figure 125:Blending of the edges produced by a 7^2x5x1 neural network arbitrator and the Roberts operator results

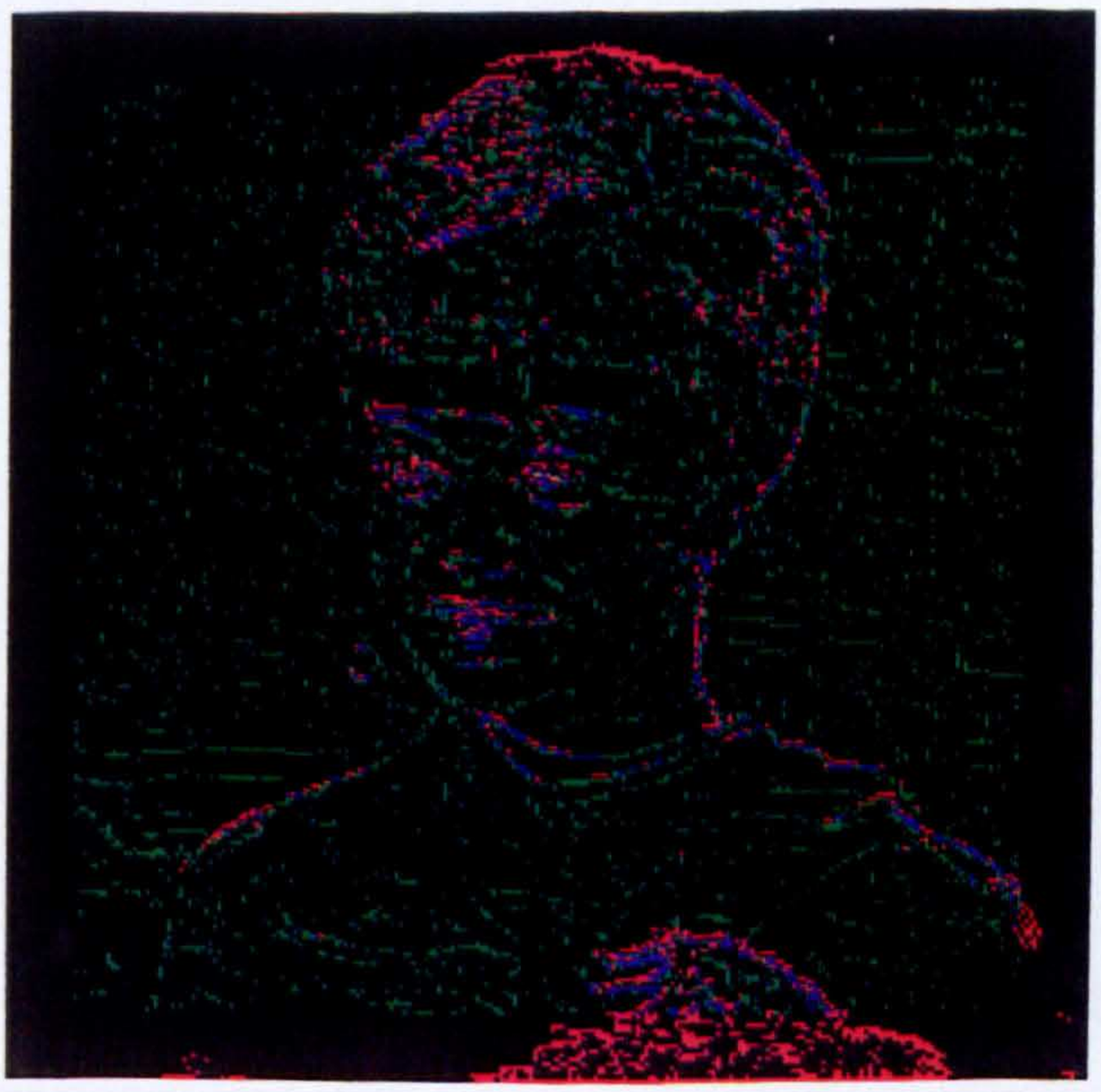

Figure 126:Blending of the edges produced by a 7^2x5x1 neural network arbitrator and the Canny operator results

Figure 127:Processed learnt image by the 5^2x30x1 neural network arbitrator

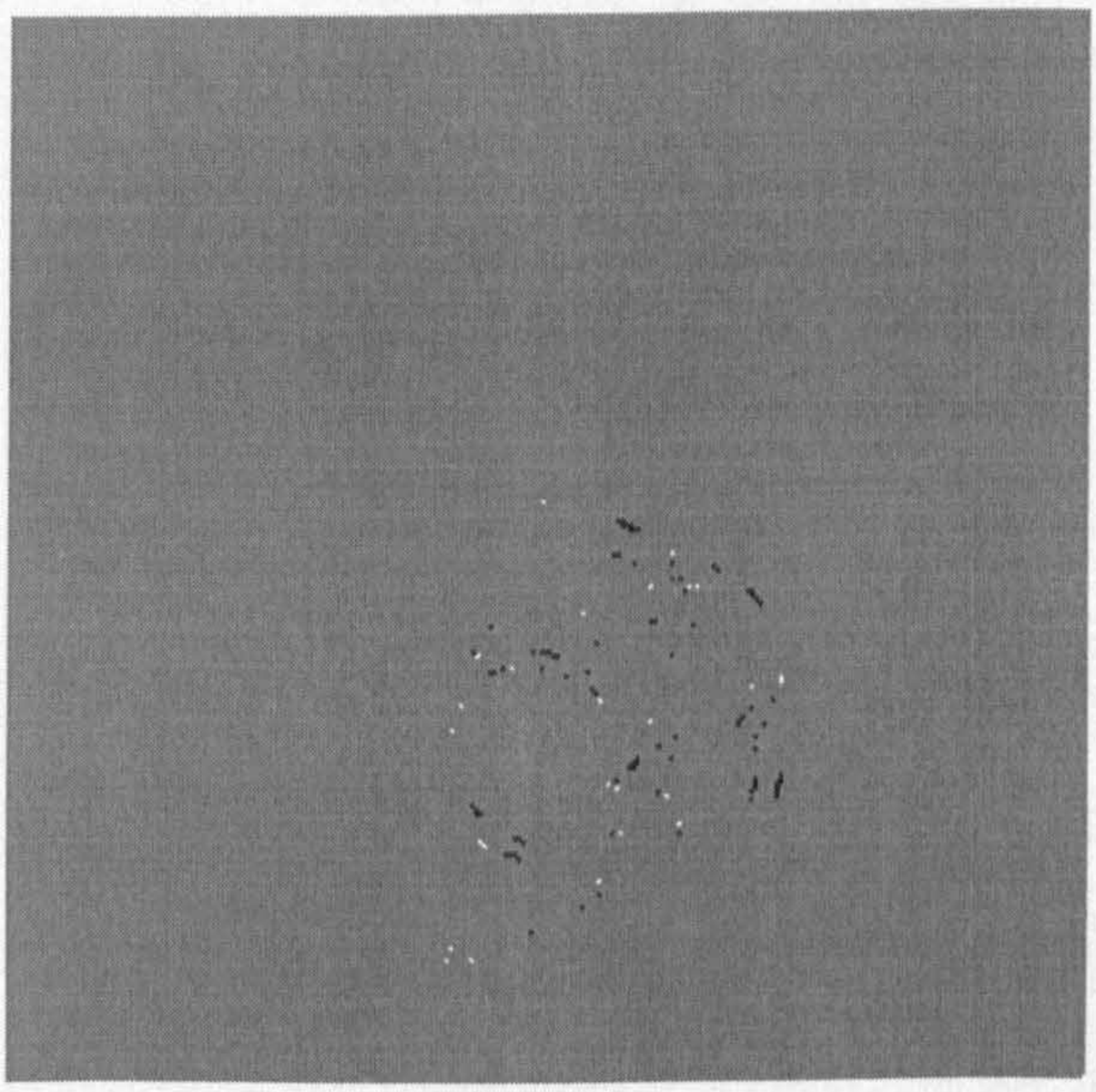

Figure 128:Comparison between Figure 127
and the reference edges

The difference image between Figure 127 and Figure 91 is presented in Figure 128. This picture has been biased at 128 grey levels, so that we can distinguish

positive and negative values.

The network performance for the test images used is presented, using the same sequence of images as before, in Figure 129 through Figure 131.

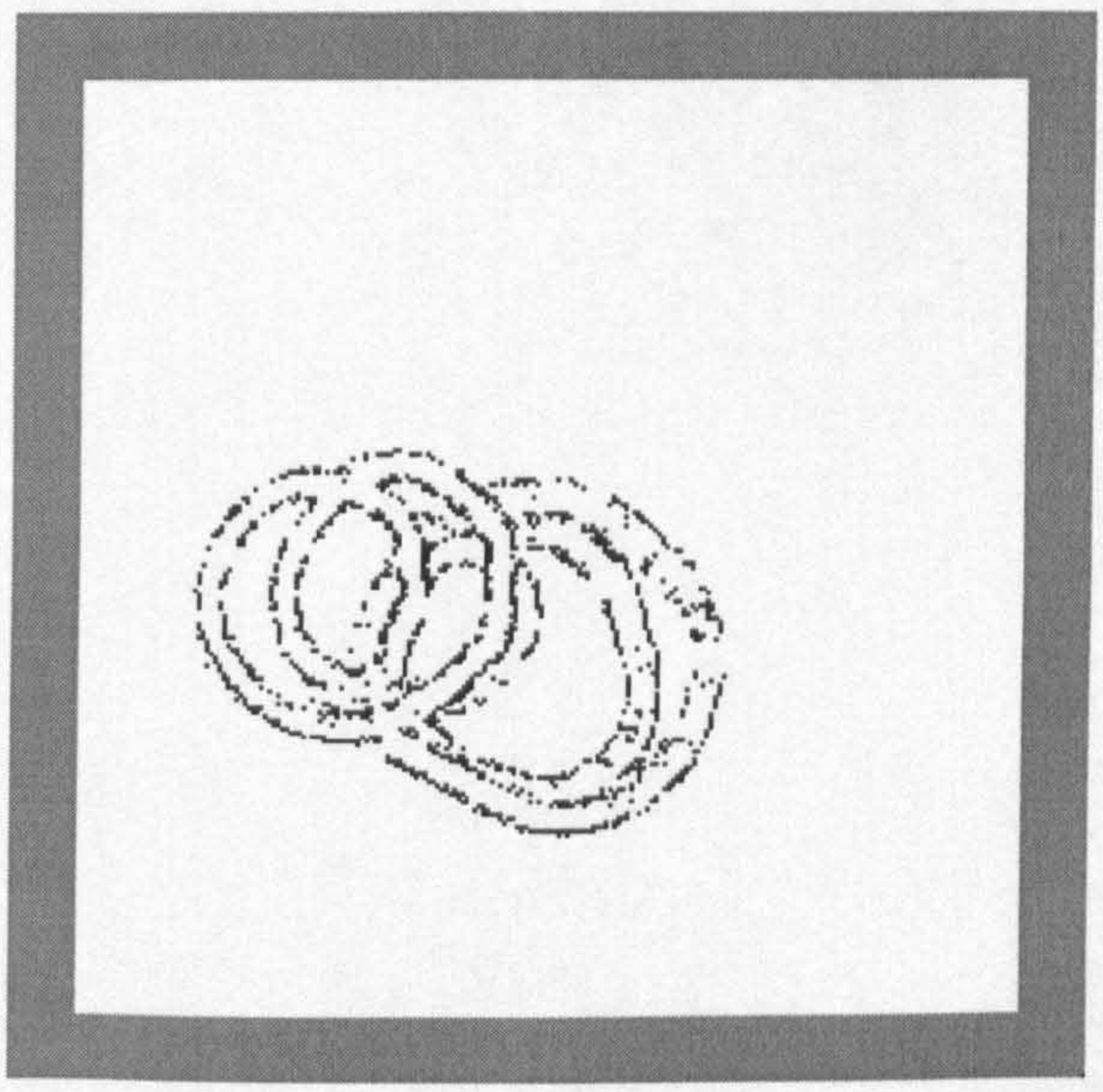

Figure 129: Band A image processed by a 5^2x30x1 network arbitrating between the Roberts and Canny edge enhanced maps

Figure 130:Lenna processed by a 5^2x30x1 network
arbitrating between the Roberts and Canny edge enhanced maps

Figure 131:Girl processed by a 5^2x30x1 network
arbitrating between The Roberts and Canny edge enhanced maps

6.5.2.5 Discussion

The first arbitration strategy investigated was between the Roberts and Canny edge detection operators. These were the *methods used throughout the* development phase. These two algorithms are particular cases: the first marks the edges rapidly and with high contrast, whilst the second marks (probably) all edges including spurious features. They were chosen as they allow for easy assessment of the arbitration strategy. Although the results can not be considered perfect (in any of the cases), the results obtained show that such a strategy is able to work and produces an edge map that both inherits characteristics from the original maps and disinherits characteristics from each of the edge enhanced maps used.

6.5.3 Case B - Roberts vs. Deriche

6.5.3.1 Methods

A second test was performed between algorithms from the same groups as Case A. The Roberts operator was maintained, and the Canny operator was replaced by a faster operator of the same group. This being the Deriche operator (section A.12). This operator was chosen as it marked fewer background edge features than the Canny operator. Effectively the edge map produced by the Deriche operator is more selective than the one produced by the Canny operator. TABLE XVIII presents the size values of the learning sets used for different window sizes. The result of processing the Band B image with the Deriche operator is presented in Figure 132. The highlighted differences between the Roberts and Deriche edge maps are presented in Figure 133. with the differences highlighted by the use of false colours. The results presented after training on a reduced learning set. are directly comparable. on a fair basis. with the ones presented in Chapter Five. For this

second test, the learning set was extracted from Band A, showing the independence of the method from the master image.

ROB+DER vs. REF	region of interest 62 x 44	
window size	learning sets	
	written	total
3x3	2,016	2728
5x5	2,289	
7x7	2,447	

Table XXXIII
Learning Sets characteristics
for Roberts vs. Deriche Arbitration

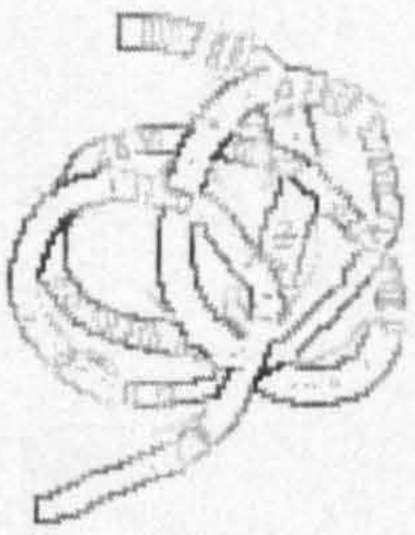

Figure 132: Band B processed by the Deriche operator

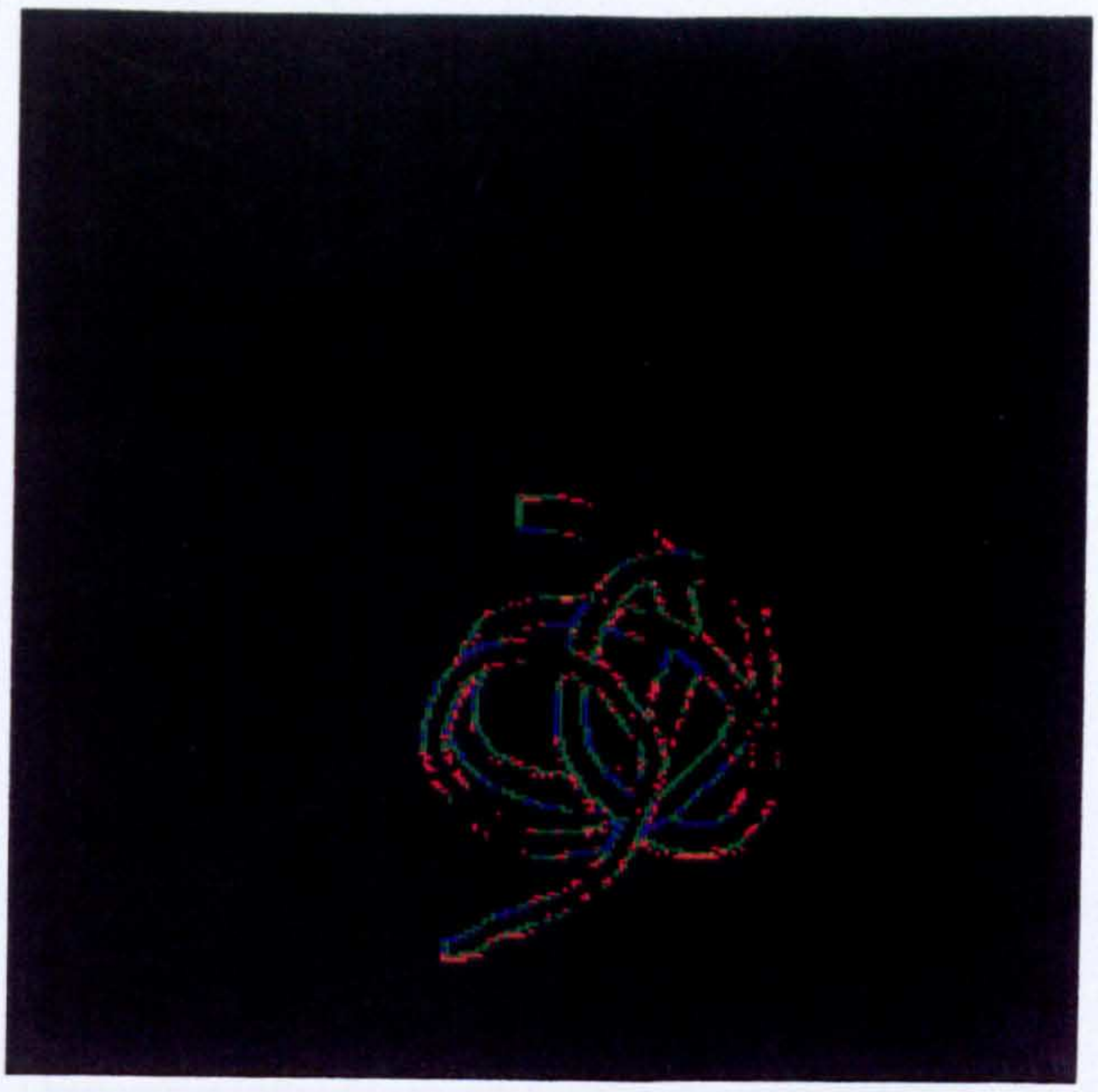

Figure 133:Highlighted differences between Figure 118 (Roberts) and Figure 132 (Deriche)

6.5.3.2 Performance

Several networks with different hidden layer size have been taught. TABLE XXXIV shows the number of unlearnt patterns and the percentage of recalled patterns achieved for these networks. Appendix C presents the performance on one of the processed images, of the networks tested for the current arbitration case. Figure 134 through Figure 136 show representative cases for each of the window sizes tested. These were selected according to quality of the edges produced. The smaller windows present the best defined detail within section E of the girl image. They also provide well defined thin edges, which are among the most continuous of all the arbitrated maps.

Performance for the Bands and Squares images are presented in TABLE XXXV and TABLE XXXVI respectively. Values are better for the Band A image as learning sets were extracted from this image. Performance is better for the smaller networks

TABLE XXXIV
CASE B
Final Values obtained for neural network arbitration between the Roberts and Deriche edge maps

Window Size 3x3	Final Values					Detail E
Hidden Nodes	WP	RCD	%	@	RMS	
10	109	1,907	94.59	1,750	0.05	21
15	110	1,906	94.54	750	0.05	28
18	90	1,926	95.54	850	0.05	9
20	99	1,917	95.09	800	0.05	14
22	101	1,915	94.99	800	0.05	24
30	81	1,935	95.98	900	0.04	25
40	103	1,913	94.89	750	0.05	24
50	68	1,948	96.63	950	0.03	26

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXXIV - A

Window Size 5x5	Final Values					Detail E
Hidden Nodes	WP	RCD	%	@	RMS	
40	40	2,249	98.25	4,850	0.02	29
50	32	2,257	98.6	5,000	0.01	17
70	29	2,260	98.73	4,800	0.01	18

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXXIV - B

Window Size 7x7	Final Values					Detail E
Hidden Nodes	WP	RCD	%	@	RMS	
20	19	2,428	99.22	4,200	0.01	31
30	19	2,428	99.22	3,300	0.01	36
40	13	2,434	99.47	5,000	0.01	32
60	15	2,432	99.39	1,880	0.01	28

WP - Number of wrongly recalled patterns @ - Iterations for convergence
RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXXIV -C

TABLE XXXV
CASE B
Performance of the arbitration process between the Roberts and Deriche edge maps when applied the band images

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3x3	10	0.48	0.3	0.4	0.28
	15	0.52	0.32	0.44	0.29
	18	0.48	0.3	0.4	0.29
	20	0.45	0.26	0.39	0.25
	22	0.46	0.3	0.31	0.3
	30	0.48	0.32	0.39	0.28
	40	0.51	0.32	0.42	0.29
	50	0.48	0.28	0.44	0.28

EQ -Edge Quality MQ - Map Quality

TABLE XXXV - A

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5x5	40	0.46	0.29	0.4	0.28
	50	0.51	0.28	0.42	0.26
	60	0.49	0.29	0.42	0.25
	70	0.49	0.27	0.4	0.23

EQ -Edge Quality MQ - Map Quality

TABLE XXXV - B

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
7x7	20	0.38	0.22	0.33	0.21
	30	0.4	0.21	0.36	0.22
	40	0.41	0.24	0.35	0.23
	50	0.37	0.2	0.34	0.22
	60	0.38	0.18	0.35	0.22

EQ -Edge Quality MQ - Map Quality

TABLE XXXV - C

TABLE XXXVI

Case B
Performance of arbitration process between the Roberts and Deriche edge maps when applied to the Squares image

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3x3	10	0.69	0.24	0.07	0.08	0.9	0.1
	15	0.65	0.22	0.6	0.08	0.9	0.1
	18	0.77	0.29	0.8	0.09	0.11	0.1
	20	0.66	0.27	0.8	0.09	0.1	0.1
	22	1.06*	0.34	0.12	0.14	0.18	0.18
	30	0.78	0.3	0.12	0.13	0.16	0.15
	40	0.78	0.35	0.9	0.12	0.13	0.14
	50	0.69	0.25	0.5	0.05	0.07	0.06

$\bar{x}$ - Average value σ - Standard deviation

* Two pixel wide vertical edges.

TABLE XXXVI - A

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
5x5	40	0.76	0.3	0.14	0.12	0.18	0.15
	50	0.67	0.28	0.1	0.11	0.13	0.13
	70	0.5	0.17	0.08	0.07	0.08	0.07

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXXVI - B

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
7x7	20	0.59	0.46	0.17	0.16	0.23	0.2
	30	0.43	0.48	0.19	0.15	0.21	0.19
	40	0.65	0.28	0.12	0.11	0.15	0.14
	50	0.59	0.41	0.17	0.13	0.19	0.17
	60	0.44	0.39	0.14	0.11	0.16	0.15

$\bar{x}$ - Average value σ - Standard deviation

TABLE XXXVI -C

Figure 134:The Girl image edge map arbitrated between the Roberts and Deriche edge maps using a 3^2x15x1 network

Figure 135:The Girl image edge map arbitrated between the Roberts and Deriche edge maps using a 5^2x50x1 network

Figure 136:The Girl image edge map arbitrated between the Roberts and Deriche edge maps using a 7^2x40x1 network

As in the previous case the lines are in the correct position and clearly reflect the edge shape. This fact is expected as the arbitration process is being performed between correct edge positions.

It has been found, as already verified for Case A (Roberts vs. Canny) edge maps, that networks working with the smaller windows present a better edge map, as the lines are thinner and they leave less points scattered around.

6.5.3.3 Discussion

Although the overall visual aspect of the presented solutions are better than Case A (Roberts vs. Deriche), the amount of work performed by the arbitration process is smaller, as the edges within the edge maps used for the arbitration process are stronger and more defined than in Case A.

Another interesting point, common with the first case presented is the fact that the best solutions seem to be achieved with smaller networks.

6.5.4 Case C - Prewitt vs. Canny

6.5.4.1 Methods

Together with the Marr and Hildreth operator the Canny operator marks all edges (probably) and is capable of detecting even poor contrast edges. This is not the case with the other operators investigated. The results previously presented using the Canny operator seem to reflect the fact that the Roberts operator did not mark low contrast edges with a strong intensity. To try to overcome this problem another test was performed. The Roberts operator was substituted by an operator with stronger marking capabilities, the Prewitt edge enhancement mask.

PWT+CAN vs. REF		
window size	learning sets	
	written	total
3x3	5,780	9240
5x5	6,463	

TABLE XXXVII

Learning sets sizes for Prewitt vs. Canny Arbitration

The learning sets used to train the neural network arbitrator were extracted from the edge map presented in Figure 137 together with the map presented in Figure 119 (Canny operator) when compared to the reference map presented

in Figure 91. The learning sets sizes used are presented in TABLE XXXVII.

Figure 137: Band B image processed by the Prewitt operator

The highlighted differences, by the use of false colours, is presented in Figure 138. Points present in both maps are represented as Green. Edges uniquely marked by the Canny edge detection operator are presented in Red. Edges uniquely marked by the operator due to Prewitt are presented in blue.

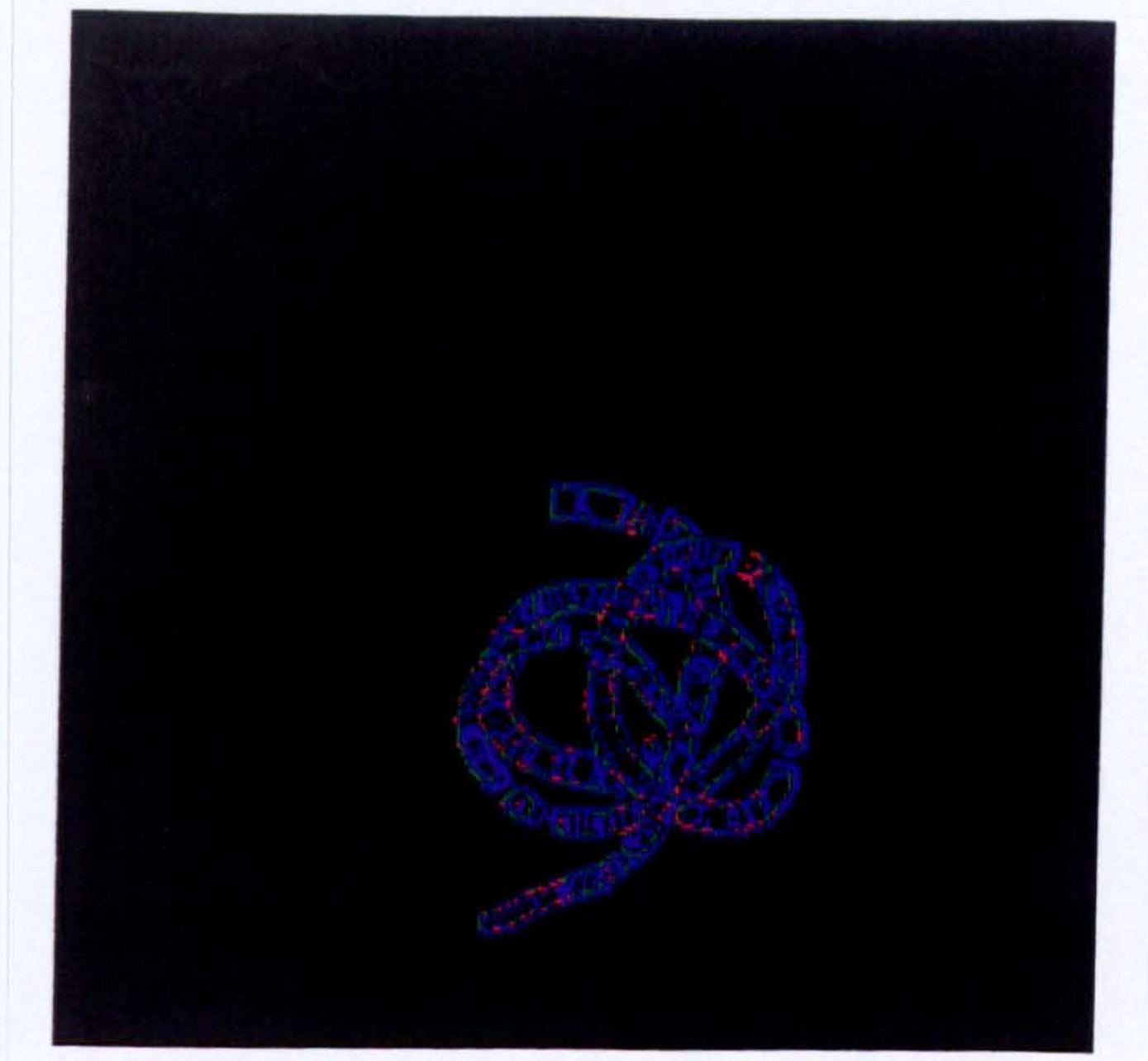

Figure 138: Highlighted differences between Figure 119 (Canny) and Figure 137 (Prewitt)

TABLE XXXVIII
CASE C
Final Values obtained for the neural network arbitrator between Prewitt and Canny edge maps

Window Size	Final Values					Detail
3x3				5,780		
Hidden Nodes	W.P.	RCD	%	@	RMS	E
5	946	4,834	83.63	1,050	0.11	19
10	635	5,145	89.01	1,050	0.09	25
15	626	5,154	89.17	1,050	0.09	33
20	482	5,298	91.66	650	0.07	25
25	476	5,304	91.76	650	0.07	32

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

TABLE XXXVIII - A

Window Size	Final Values					Detail E
5x5				6,463		
Hidden Nodes	W.P.	RCD	%	@	RMS	
15	277	6,186	4.29	200	0.04	13*
20	126	6,337	1.95	450	0.02	28
25	105	6,358	1.62	850	0.02	31
30	95	6,368	1.47	500	0.01	23
40	62	6,401	0.96	500	0.01	--**
50	98	6,365	1.52	500	0.04	26

WP - Number of wrongly recalled patterns @ - Iterations for convergence

RCD - Number of correctly recalled patterns RMS - Root mean square error at convergence

* Difficult to distinguish

** Indistinguishable

TABLE XXXVIII - B

TABLE XXXIX

CASE C

Performance of the arbitration process between Prewitt and Canny edge maps when applied to the Band images

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3x3	5	0.6	0.31	0.64	0.46
	10	0.55	0.37	0.71	0.53
	15	0.61	0.36	0.76	0.58
	20	0.48	0.34	0.79	0.59
	25	0.55	0.39	0.8	0.61

EQ -Edge Quality MQ - Map Quality

TABLE XXXIX - A

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5x5	15	0.52	0.33	0.85	0.62
	20	0.52	0.38	0.97	0.9
	25	0.55	0.39	0.98	0.92
	30	0.52	0.35	0.98	0.93
	40	0.57	0.4	0.99	0.95
	50	0.55	0.38	0.99	0.94

EQ -Edge Quality MQ - Map Quality

TABLE XXXIX - B

TABLE XL
CASE C
Performance of the arbitration process between Prewitt and Canny edge maps when applied to the Squares image

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3x3	5	1.07	0.6	0	0	0.01	0.01
	10	0.94	0.43	0.02	0.03	0.05	0.04
	15	1.13	0.5	0.02	0.02	0.06	0.3
	20	1.24	0.34	0.08	0.06	0.19	0.11
	25	0.93	0.33	0.12	0.08	0.25	0.12

$\bar{x}$ - Average value σ - Standard deviation

TABLE XL - A

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
5x5	15	0.89	0.17	0.08	0.05	0.14	0.04
	20	1.13	0.23	0.11	0.5	0.21	0.4
	25	0.99	0.31	0.07	0.04	0.14	0.5
	30	0.97	0.2	0.06	0.04	0.12	0.05
	40	1	0.25	0.09	0.06	0.17	0.07
	50	1.06	0.32	0.12	0.07	0.22	0.08

$\bar{x}$ - Average value σ - Standard deviation

TABLE XL - B

6.5.4.2 Performance

In previous cases larger windows did not present an increase in the performance of the networks. Due to this reason 7x7 windows were not tested. TABLE XXXVIII presents the learning characteristics for a number of hidden nodes. TABLE XXIX and TABLE XL present the performance for the Band and Squares images. A selection of the best performing networks is easy to obtain if we look at the different "Girl" pictures. Effectively, as in the cases

before, the smaller networks (3^2 x 5 x 1) converged to present better definition of the lines, even in completeness (note the left shoulder in the girl image, see Figure 139), while removing the majority of the artefacts in the background. A second possible choice is the solution including 15 hidden nodes (Figure 140). This solution, despite the inclusion of more background points, also includes more lines (see the upper right line corresponding to the border of the window, or the line between neck and jaw in Figure 139).

Figure 139 :Girl image arbitrated between the Prewitt and Canny edge maps using a 3^3 x 5 x 1 network

Figure 140 :Girl image arbitrated between the Prewitt and Canny edge maps with a 3^3 x 15 x 1 network

Solutions obtained when using 5^2 windows are not very good with weak edges of an ill defined shape being produced and a large amount of scattered points in the background.

6.5.4.3 Discussion

A comparison between the case reported in this section and case A . where the Roberts and Canny operators were used. shows that a more complete set of lines is achieved when the arbitration is done between the Prewitt and Canny edge detectors. This is expected as Prewitt edge maps mark the edges with more intensity than the Roberts operator. Not only is the Detail E more defined in this case. where the roundness is clearly visible. but difference can also be seen on the left side of the girls face. Resulting edge maps present a similar aspect as in case A. presented before.

6.5.5 CASE D - Roberts vs. Prewitt

6.5.5.1 Methods

A fourth test was performed. In the previous cases, different approaches were mixed. In this case we will arbitrate two similar approaches. Advantages are expected in the form of a reduction in the high intensity features from the background, which are due to noise or other causes. In this case we are leaving the arbitration process to operate mainly on the lines. However we are loosing one component of the information present in the previous cases, which corresponds to the thickness of the edges. Images used for the learning set extraction were presented in Figure 118 and Figure 137 .

A blending of these two images is presented in Figure 141. These are thresholded to a similar number of marked points (10% of the whole image). The similarity of the edge maps originates the need to threshold. In the previous presented cases there are enough meaningful differences to allow a wide choice for the threshold. before blending, without affecting the analysis of the blended image. In this case as the width of the edges, which can be extracted from the Prewitt edge map, includes the vast majority of the points marked by the Roberts operator. differences are hardly perceived. The value of the threshold. is imposed by the Roberts operator (th=1), in order to obtain the biggest number of marked edges. A similar percentage of marked points was chosen for the Prewitt edge map image, in order to achieve a similar comparison basis. The images are blended in the same ratios as used before (1/3 and 2/3), and the LUT changed to false colours. Green assigned to points marked only by the Prewitt operator. Red to points marked only by the Roberts operator. and Blue for common points. Table XLI gives the learning set characteristics and table XLII gives the results of the arbitration.

ROB+PWT vs. REF	region of interest 84 x 110	
window size	learning sets	
	written	total
3x3	5,879	9240
5x5	5,527	

TABLE XLI
Learning set characteristics

The similarity of the edge maps does not allow for a large area to be arbitrated upon. The main objective of the work in this case is to evaluate the response in the presence of large and undefined edge positions.

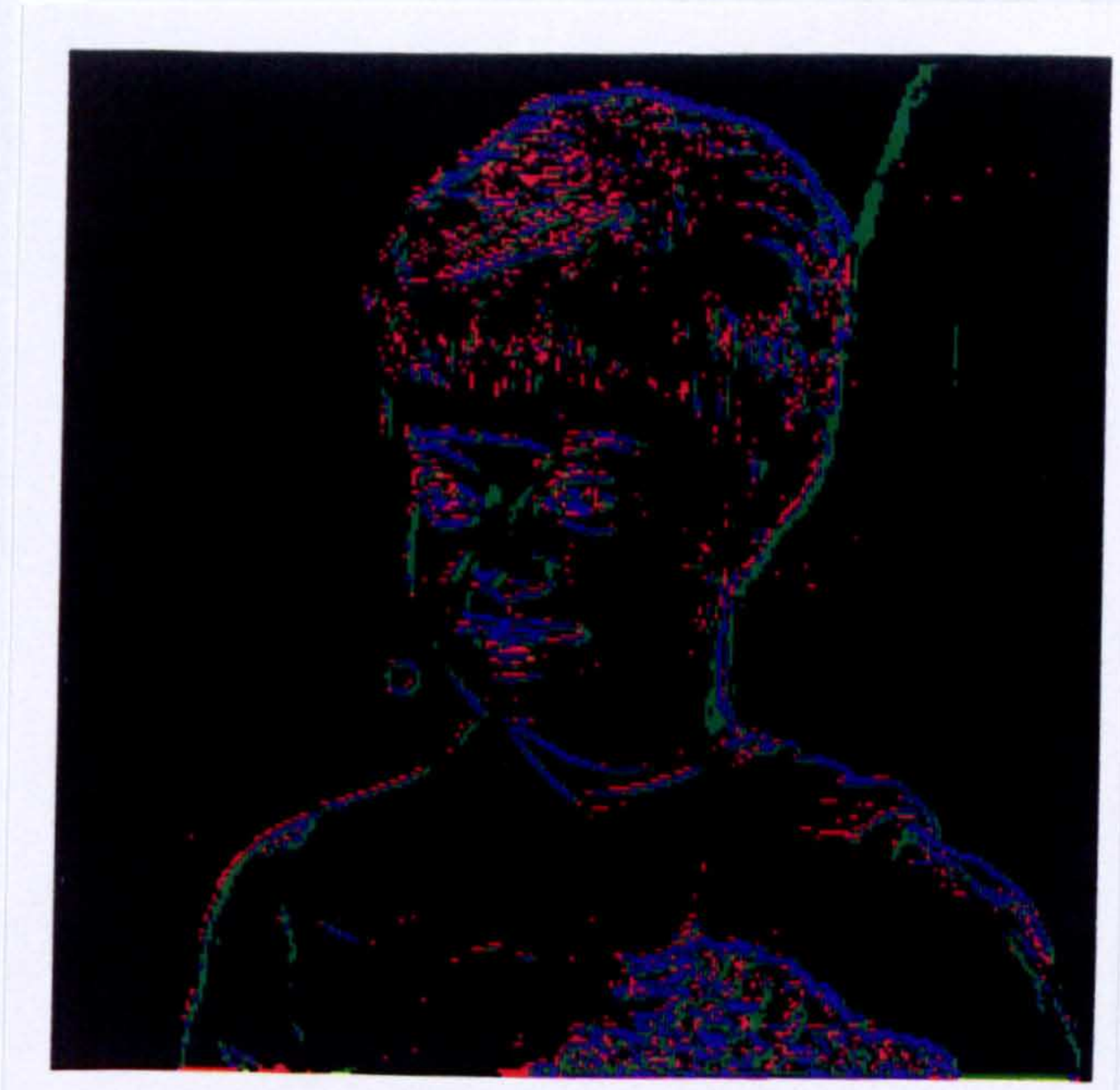

Figure 141:Blending of the Girl image processed by the Roberts and Prewitt operators thresholded to 10% of marked points

TABLE XLII
Case D
Final Values obtained for the neural network arbitrator
between the Roberts and Prewitt edge maps

Window Size	Final Values					Detail E
3x3						
Hidden Nodes	W.P.	RCD	%	@	RMS	
5	704	5,175	88.03	800	0.09	29
10	547	5,332	90.7	700	0.08	20
15	492	5,387	91.63	650	0.07	25
20	369	5,510	93.72	650	0.06	23
25	372	5,507	93.67	500	0.05	22
30	323	5,556	94.51	550	0.05	24
WP - Number of wrongly recalled patterns			@ - Iterations for convergence			
RCD - Number of correctly recalled patterns			RMS - Root mean square error at convergence			

TABLE XLII - A

Window Size	Final Values					Detail E
5x5						
Hidden Nodes	W.P.	RCD	%	@	RMS	
10	403	6,124	93.83	200	0.05	36
15	251	6,276	96.15	250	0.03	38
20	137	6,390	97.9	250	0.02	39
WP - Number of wrongly recalled patterns			@ - Iterations for convergence			
RCD - Number of correctly recalled patterns			RMS - Root mean square error at convergence			

TABLE XLII - B

6.5.5.2 Performance

Figure 142 presents one of the solutions of the arbitrated picture superimposed on the original girl image. The smallest network is presented, as it presents the least number of background points marked (two in the window and one in the left of the shoulder). It also presents the best edge definition in terms of width. as can be seen in the left shoulder line. TABLE XLIII and TABLE XLIV present the performance for the Band and Squares images.

Figure 142: Superposition of the processed image by a 3^2 x 5 x 1 network, onto the original girl image

TABLE XLIII
Case D
Performance of the arbitration process between the Roberts and Prewitt edge maps when applied to the Band Images

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
3x3	5	0.71	0.46	0.73	0.56
	10	0.71	0.47	0.8	0.61
	15	0.61	0.42	0.73	0.61
	20	0.6	0.44	0.73	0.65
	25	0.67	0.45	0.84	0.68
	30	0.66	0.41	0.83	0.67

EQ -Edge Quality MQ - Map Quality

TABLE XLIII - A

Window Size	*Hidden Nodes*	Band A Image		Band B Image	
		EQ	MQ	EQ	MQ
5x5	10	0.61	0.45	0.78	0.68
	15	0.68	0.46	0.85	0.74
	20	0.66	0.5	0.87	0.78

EQ -Edge Quality MQ - Map Quality

TABLE XLIII - B
TABLE XLIV
CASE D
Performance of the arbitration process between the Roberts and Prewitt edge maps when applied to the Squares image

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
3	5	1.17	0.71	0	0	0	0
	10	0.96	0.58	0.01	0.01	0.02	0.01
	15	0.98	0.58	0.02	0.02	0.03	0.03
	20	1.03	0.49	0.02	0.03	0.03	0.03
	25	1.08	0.56	0.01	0.02	0.02	0.02
	30	0.82	0.57	0.01	0.02	0.02	0.02

$\bar{x}$ - Average value σ - Standard deviation

TABLE XLIV - A

Window Size	*Hidden Nodes*	Lines		Inside		Outside	
		$\bar{x}$	σ	$\bar{x}$	σ	$\bar{x}$	σ
5	5	1.26	0.55	0.1	0.1	0.19	0.17
	10	0.97	0.37	0.03	0.02	0.04	0.03
	30	1.07	0.46	0.02	0.02	0.03	0.03

$\bar{x}$ - Average value σ - Standard deviation

TABLE XLIV - B

6.5.5.3 Discussion

This last example. although it produces the clearest map with the fewest scattered points. does not represent the highest achievement from the

presented examples. The edge enhanced maps being arbitrated upon are very similar and the edge detection algorithms chosen are more selective in marking the sharpest edges than the others investigated.

The arbitration scheme produces a more complete map than that produced by the Roberts operator. However, the arbitration scheme produces a similar result to the edge map produced by the Prewitt Operator with a suitable threshold. For the same threshold condition as used to produce the image in Figure 141, a comparison between the edge map produced by the Prewitt operator and the edge map that was superimposed on to Figure 142, shows that a gain of 131 points was obtained relative to the Prewitt edge map, and a loss of 1099 points also occurred. This value is higher than the number of retained points (1089). There was no improvement in the definition of the edges, as a decrease in edge width was not achieved. Although this case has produced the best edge maps, it is as a direct result of the data fed into the network. and not due to an outstanding performance of the arbitration system.

6.5.6 Closure

In this section several edge detection arbitration schemes have been presented. These were presented along with an analysis of their performance and efficiency. The reasons that suggested the tuples to be chosen were presented as were the comparisons performed and measurements taken.

Comparisons between the different edge detection arbitration schemes are next to be described. These will be based on the measures presented earlier. Finally conclusions will be drawn.

6.6 Comparisons

6.6.1 Foreword

In the previous section we have presented several examples of arbitration schemes along with representative examples of their performance. In the remainder of this chapter. comparisons between the proposed solutions will be presented. The comparisons presented are the best example for each particular case. Finally global comparisons will be drawn.

6.6.2 Computational Load

The arbitration process is more computational intensive than the methods arbitrated upon. Also. as more than one window is used it requires more calculations than the equivalent neural network edge detector using the same window size. Although these factors are drawbacks to practical application of the system. they can be overcome using parallel processing techniques. described in Chapter Four.

6.6.3 Convergence of the solutions

Method		Window Size	Hidden Nodes	%	MQ_B	EQ_B	Line ($\bar{x}$)	σ
Neural Network Edge Detector		3	40	91.81	0.84	0.07	0.94	0.34
		5	50	95.83	0.95	0.81	0.8	0.32
Techniques Arbitrated:								
Roberts	Canny	3	50	95.17	0.56	0.4	0.94	0.42
		5	50	98.41	0.98	0.91	0.95	0.34
		7	50	99.22	0.98	0.92	0.99	0.25
Roberts	Deriche	3	30	95.98	0.39	0.28	0.78	0.3
		5	70	98.73	0.4	0.23	0.64	0.16
		7	30	99.47	0.36	0.22	0.55	0.47
Prewitt	Canny	3	25	91.76	0.8	0.61	0.93	0.23
		5	40	99.04	0.99	0.95	1	0.25
Robert	Prewitt	3	30	94.51	0.83	0.67	0.38	0.02
		5	20	97.96	0.87	0.78	1.07	0.46

% - Percentage of correctly recalled Patterns
EQ_B - Edge Quality for Band B image
MQ_B - Map Quality for Band B image
$\bar{x}$ - Average value
σ - Standard deviation

TABLE XLV
Learning achievements

The first factor compared. in table XLV, is the percentage of correctly recalled patterns between the different cases. Images processed by the networks that presented the highest recalling factors, are given in figures 143 through 147. First it should be recalled that the Edge Quality Parameter and the recalling factor are distinct quantities as they refer to distinct sets of points. Values obtained show that network performance is not proportional to the percentage of recalled patterns. This fact can be seen. through the tables presented so far. as better scores did not coincide with better recalling factors. It can also be noted from tables TABLE XLVI and TABLE XLVII that the optimum number of hidden nodes for each figure of merit were normally different from the equivalent solution presented in TABLE XLV . This is due to the distinct figures of merit used throughout the work.

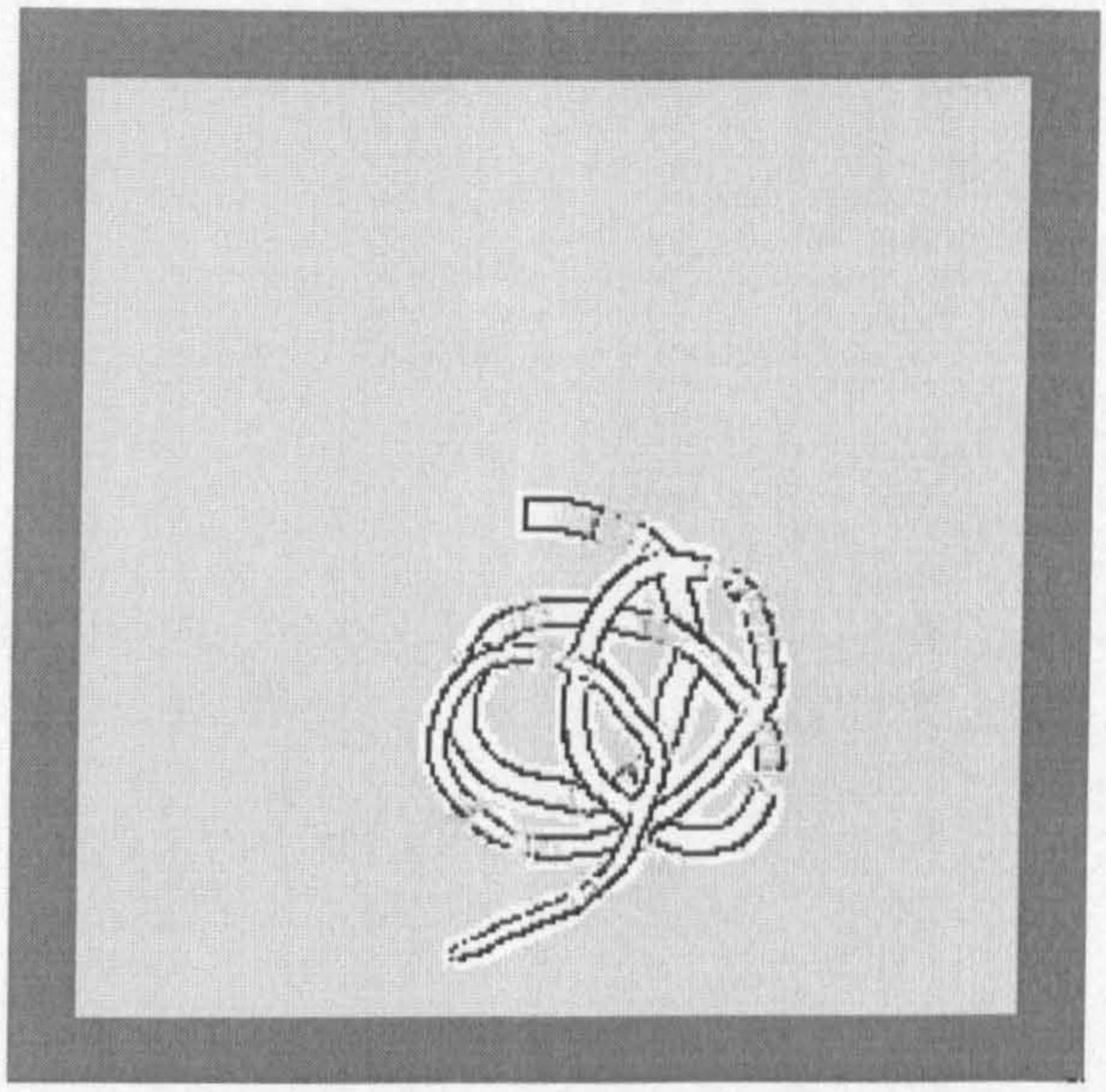

Figure 143: Edge detection of Band B using a 3^2x50x1 neural network

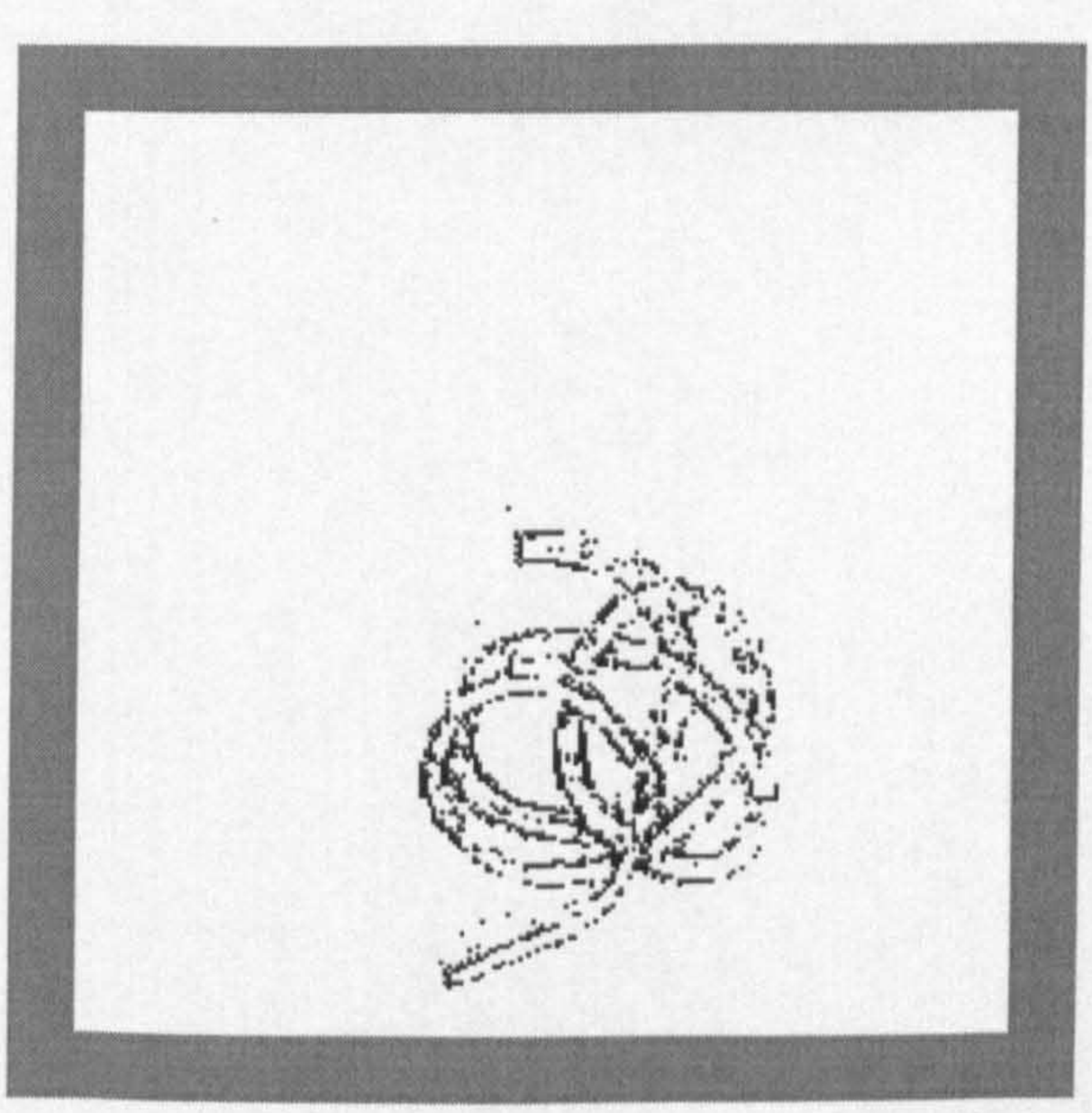

Figure 144- Band B edges arbitrated between the Roberts and Canny produced edge maps using a 3^2x50x1 neural network

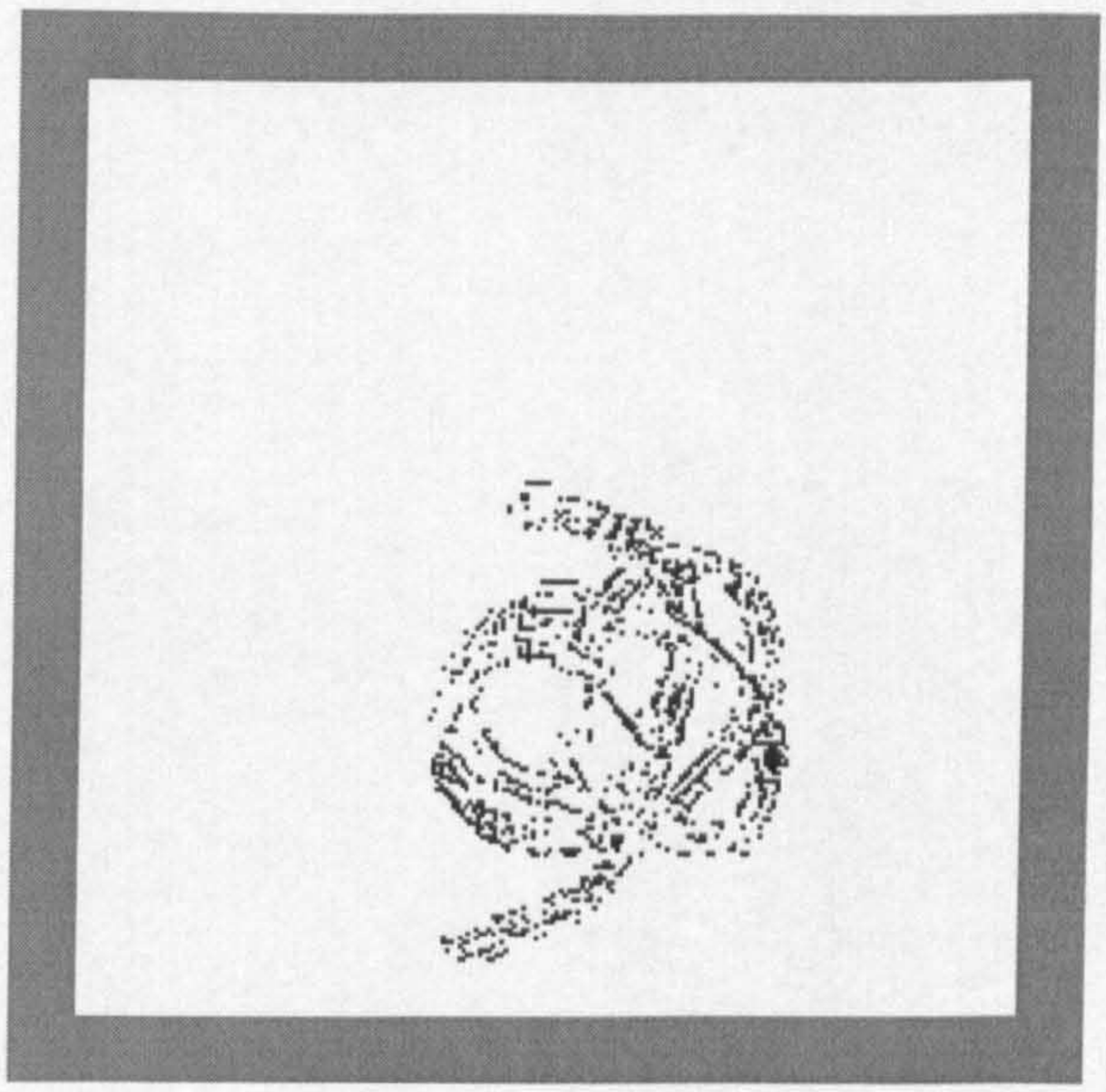

Figure 145 -Band B edges arbitrated between the Roberts and Deriche produced edge maps using a 7^2x30x1 neural network

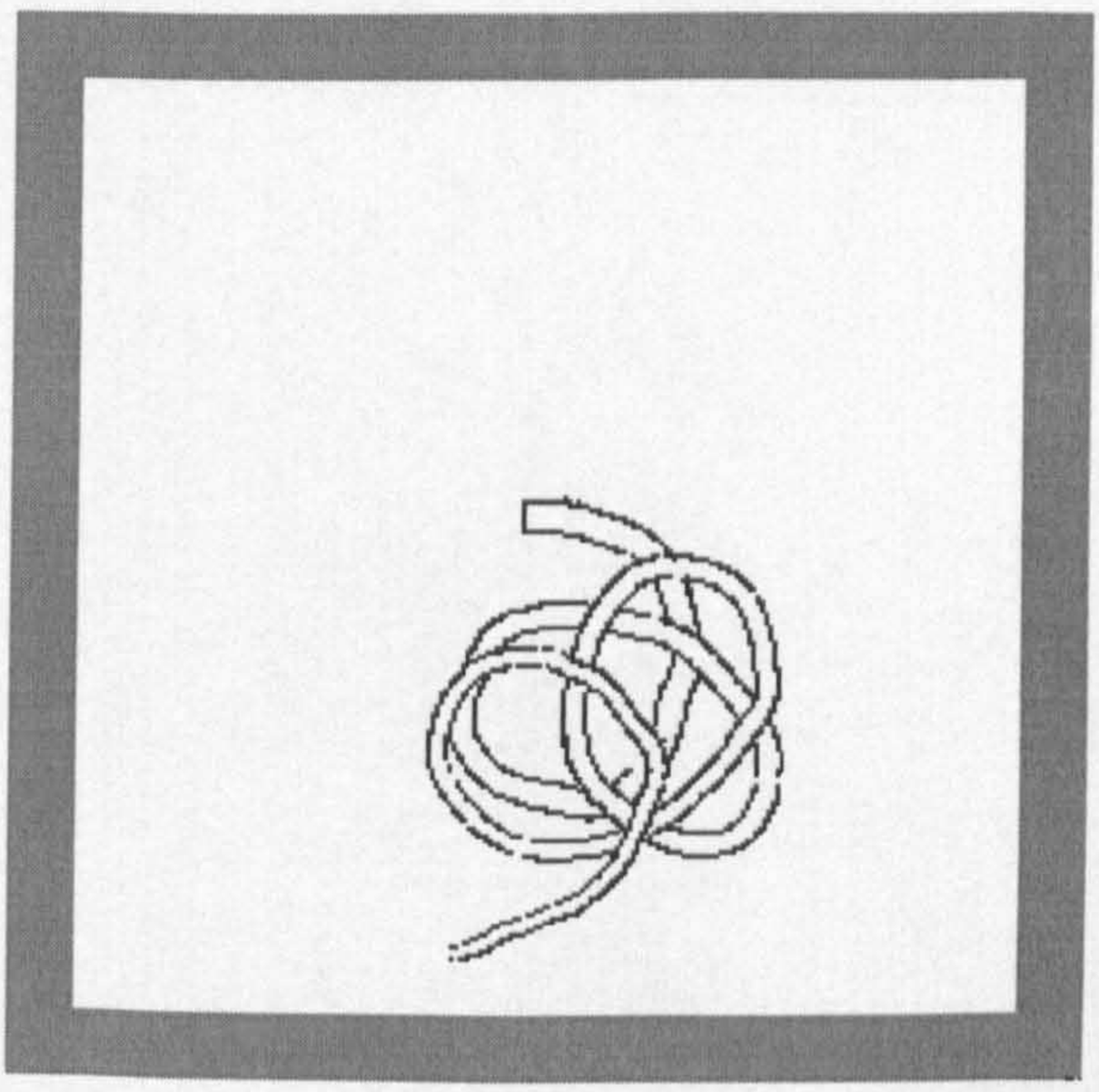

Figure 146 -Band B edges arbitrated between the Prewitt and Canny produced edge maps using a 5^2x40x1 neural network

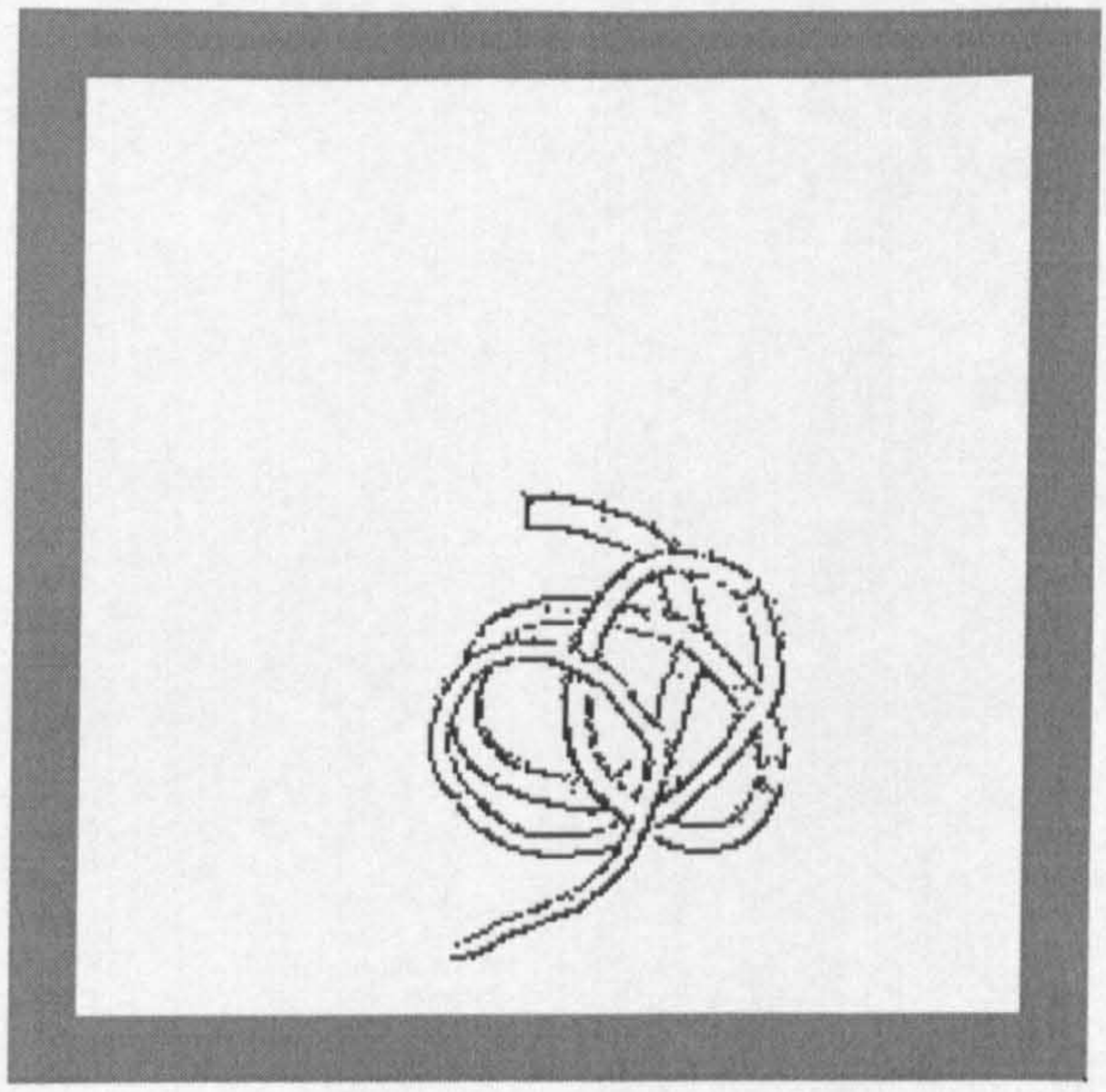

Figure 147 -Band B edges arbitrated between Roberts and Prewitt produced edge maps using a 5^2x20x1 neural network

6.6.4 Robustness

Robustness of the arbitration examples can be evaluated by a number of means. First it can be evaluated by the standard deviation presented for the various marked lines, as they are differently marked as the amount of noise varies. Secondly, and independently from the grey level, it can be evaluated by the number of points obtained outside the squares, in the case of the squares image. The best values obtained for each of the different networks are presented in TABLE XLVI. The first factor that can be noted is the closeness of the length of the average line to one. Values measured for the edge detectors presented in TABLE XLVI were generally larger than one with larger standard deviations. Values outside the square are consistently of small value showing an insensibility to noise that outperforms some of the methods arbitrated upon.

Generally values are situated between the scores presented by the methods arbitrated upon.

Method		Window Size	Hidden Nodes	Line		Outside	
				$\bar{x}$	σ	$\bar{x}$	σ
Neural Network Edge Detector		**3**	**40**	**0.94**	**0.34**	**0.06**	**0.08**
		5	50	0.8	0.32	0.01	0.01
Techniques Arbitrated							
Roberts	Canny	3	18	0.98	0.5	0.03	0.02
		5	50	0.95	0.34	0.1	0.06
		7	50	**0.99**	**0.25**	**0.08**	**0.04**
Roberts	Deriche	**3***	**30**	**0.78**	**0.3**	**0.16**	**0.15**
		5	40	0.76	0.3	0.18	0.15
		7	40	0.65	0.28	0.15	0.14
Prewitt	Canny	3	10	0.84	0.43	0.05	0.04
		5	**25**	**0.99**	**0.31**	**0.14**	**0.05**
Roberts	Prewitt	**3**	**20**	**1.03**	**0.49**	**0.03**	**0.03**
		5	10	1.26	0.55	0.19	0.17
* Not considered HN=22, Lines=1.06±0.34				$\bar{x}$ - Average value		σ - Standard deviation	

TABLE XLVI
Performance for the Squares Image
Best Values

Figure 148 through Figure 152 present the images with the best line measures for each of the cases presented (in bold in TABLE XLVI) . The best values were obtained for the arbitrated Roberts and Canny edge detector map using a 7^2x50x1 neural network, as this presents the closest to the unit value and the smallest standard deviations. However, a comparison between the pictures presented show that the value does not correspond to the best image obtained. The values obtained reflect the fact that edges are marked with a width of more than one pixel in some areas. The best solution, in visual terms, in this case seems to be that obtained by the neural network edge detector using a 3^2x40x1 network.

Several comments can be drawn from the presented images. Firstly a better edge definition is obtained from the Neural Network Edge detector in

comparison with the arbitrated maps. The images, obtained using the Neural Network edge detector, present clear edges as a continuous thin line, with correct shape and position. This fact is reflected in the values presented in TABLE XLVI.

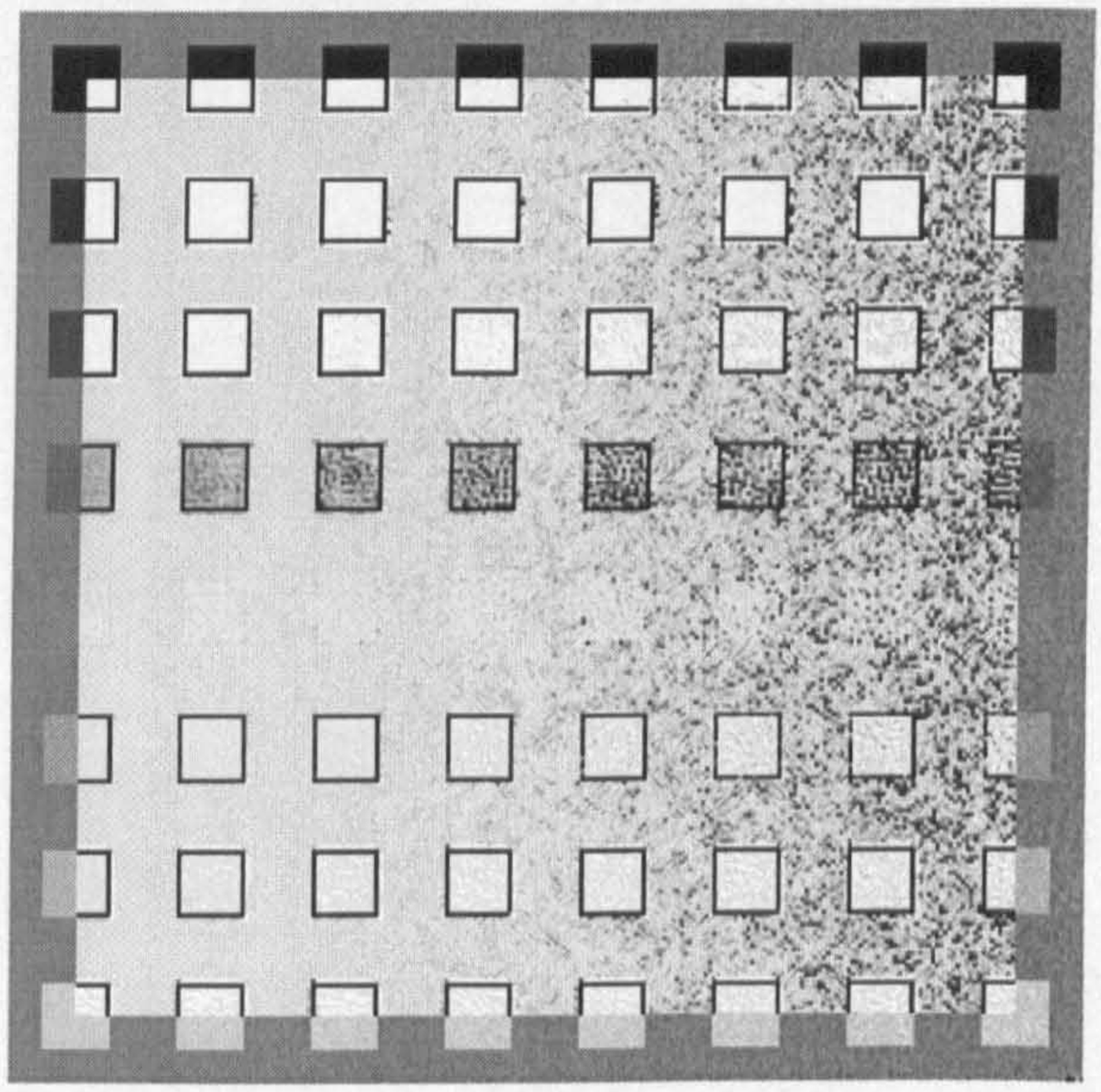

Figure 148: Neural Network Edge Detection using a 3^2x40x1 neural network

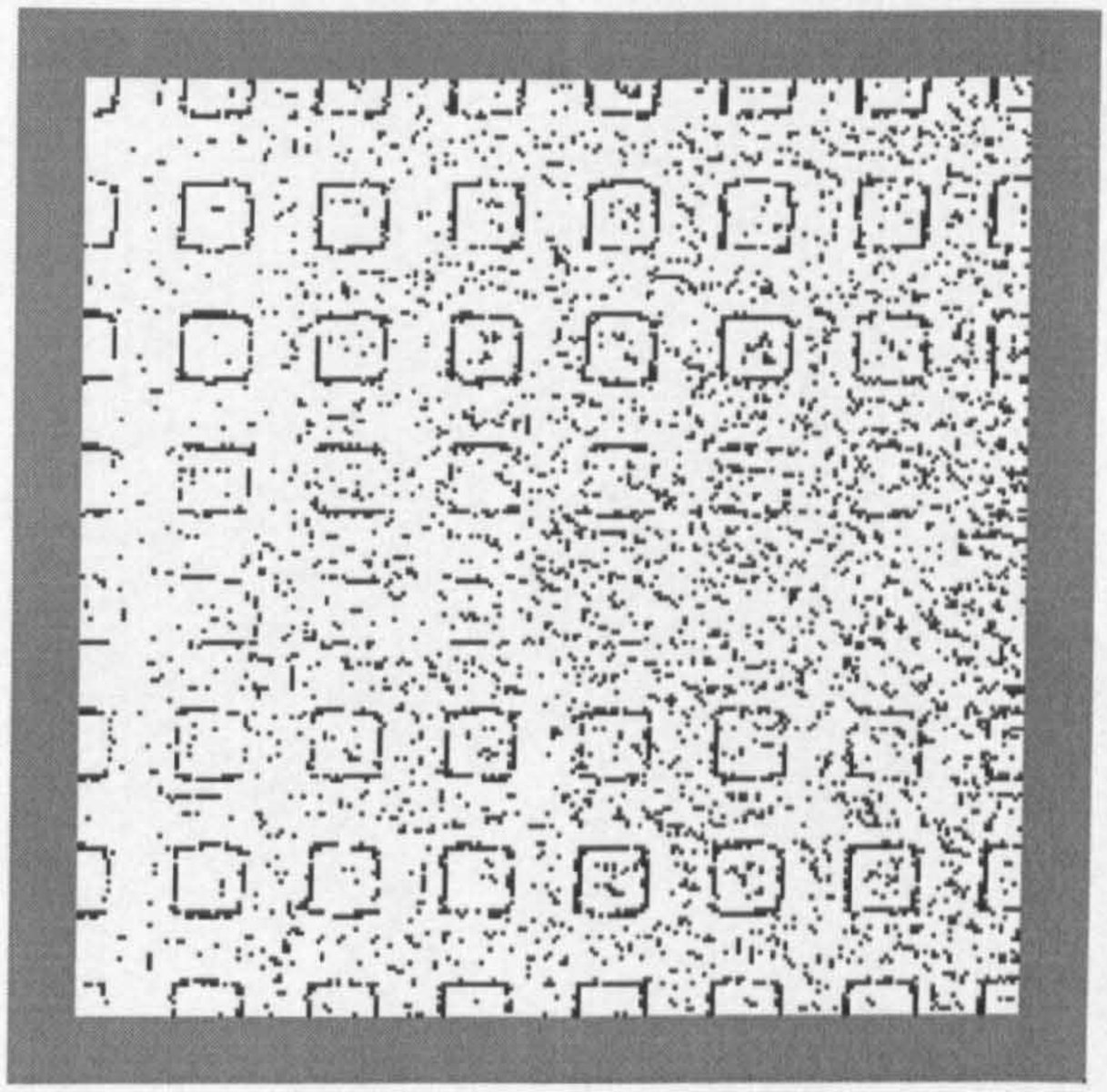

Figure 149 - Squares edges arbitrated between the Roberts and Canny produced edge maps using a 7^2x50x1 neural network

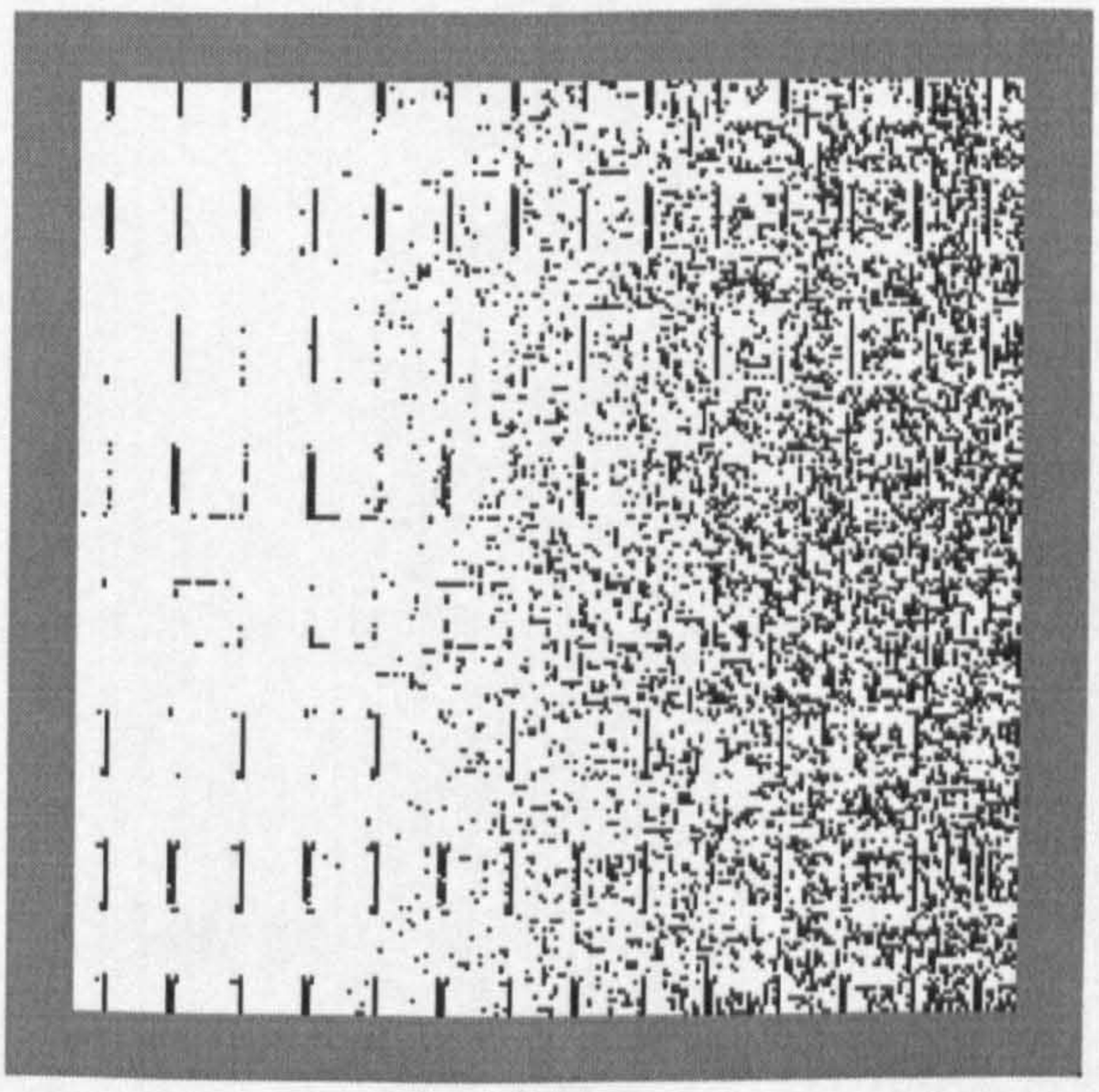

Figure 150 - Squares edges arbitrated between the Roberts and Deriche produced edge maps using a 3^2x30x1 neural network

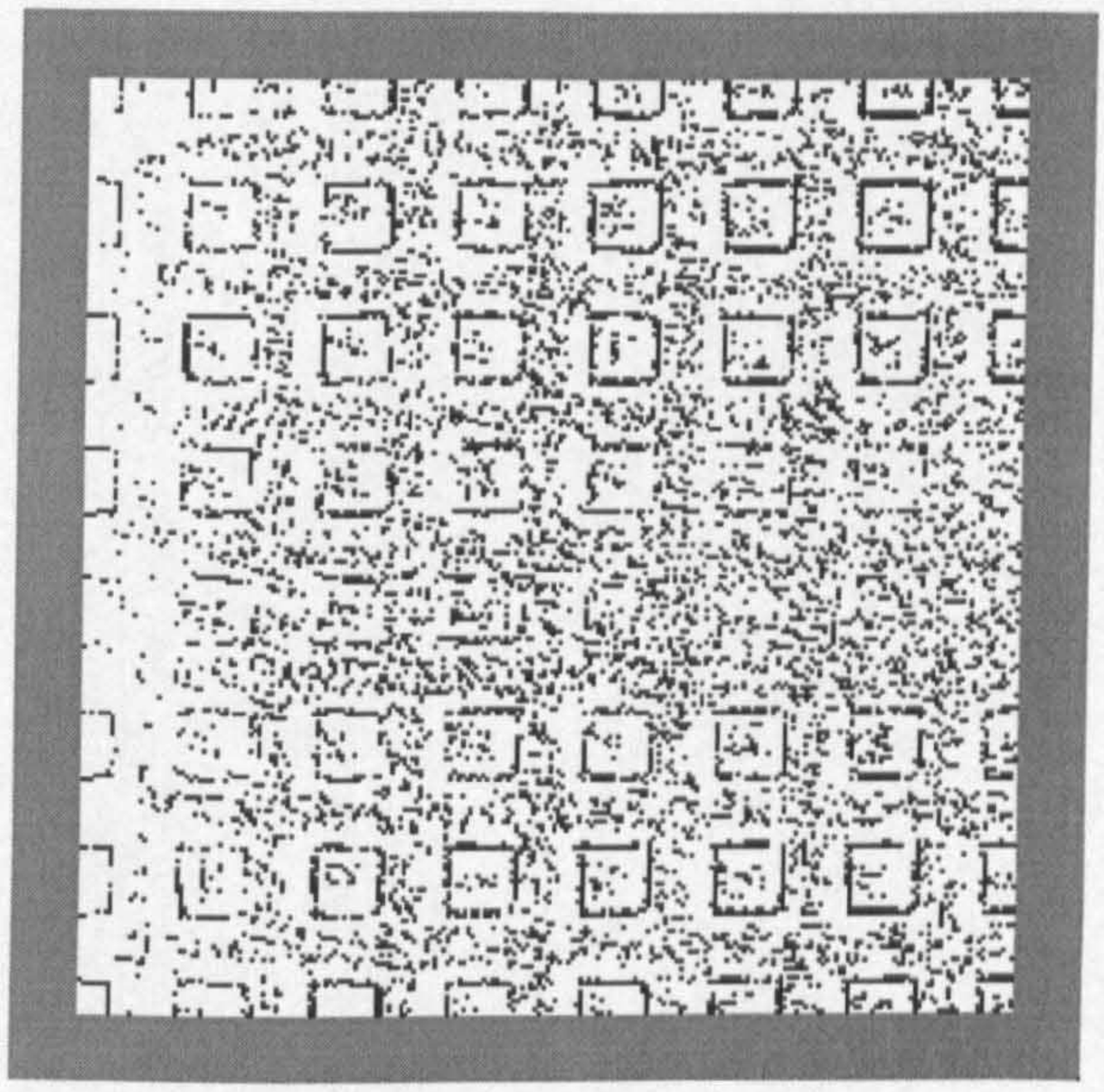

Figure 151 - Squares edges arbitrated between the Prewitt and Canny produced edge maps using a 5^2x25x1 neural network

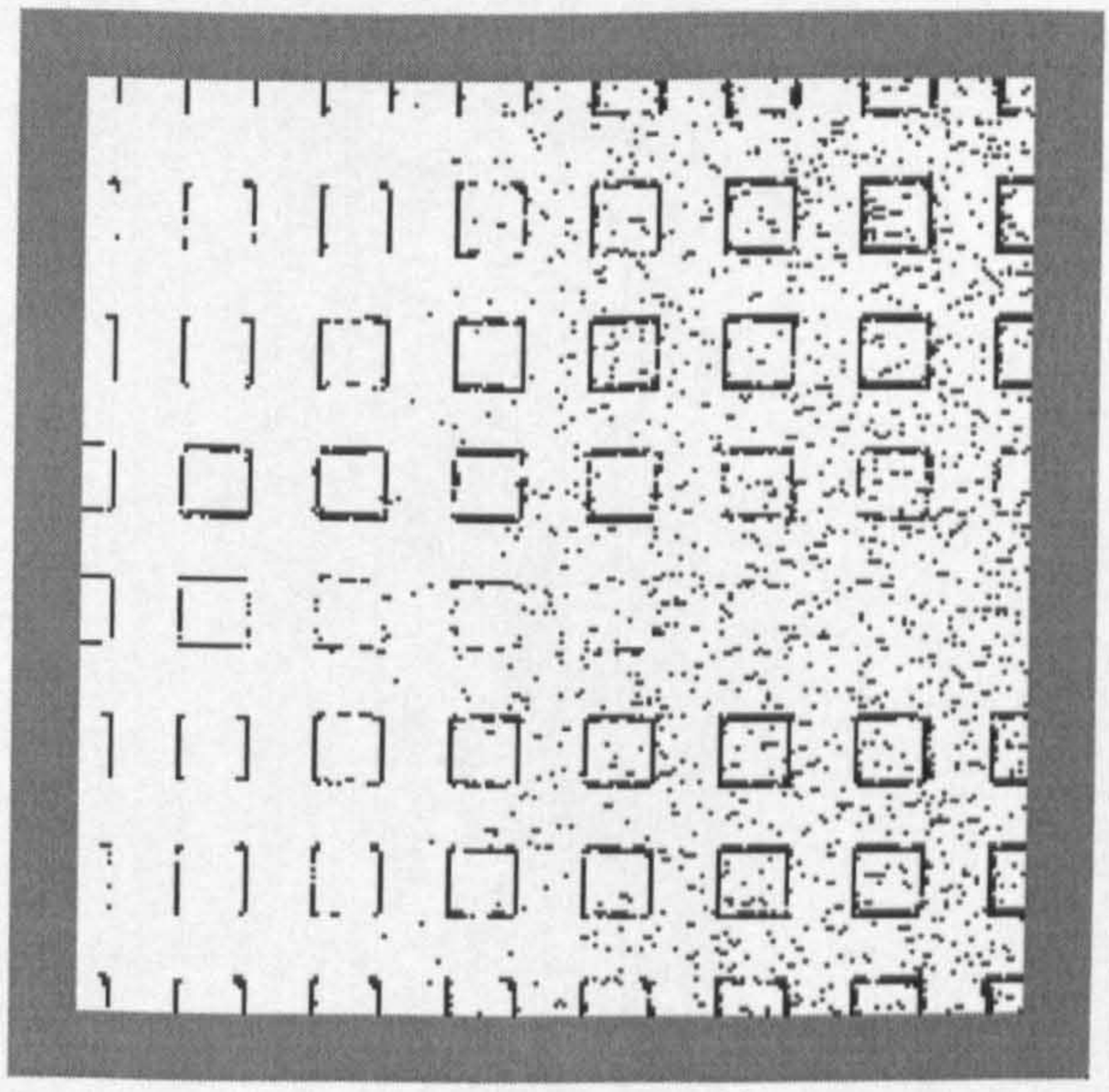

Figure 152 - Squares edges arbitrated between the Roberts and Prewitt produced edge maps using a 3^2x20x1 neural network

The second detail, that grasps the attention, is the directional behaviour present in some of the resulting maps of the arbitration system and for various edge detection neural networks. This fact suggests that a bias exists in the learning set derived from the learning image used.

Finally some of the solutions perform better for medium quantities of added noise (in terms of the square picture scale), performing poorly in the total absence of noise. The same behaviour is also observed, although less frequently, in the case of varying grey level. Intermediate contrasts are marked as edges whilst low and high contrast edges have been lost.

6.6.5 Sensibility and Overall Performance

Figures 153 to 157 present the 'Girl' image processed by the neural networks which achieved the highest Edge Map Quality scores presented in TABLE XLVII.

Method		Window Size	Hidden Nodes	MQ	
				A	B
Neural Network Edge Detector		5	50	0.61	0.81
Techniques Arbitrated:					
Roberts	Canny	5	50	0.36	0.91
Roberts	Deriche	3	15/40	0.32	0.29
Prewitt	Canny	5	40	0.4	0.95
Roberts	Prewitt	5	20	0.5	0.78

TABLE XLVII
Best Edge Map Quality
obtained by the solutions presented

Measures for Roberts and Deriche produced edge maps are superior for Band A. This is due to the fact that these networks were taught with learning sets extracted from this picture (see 6.5.3.1), instead of Band B like the other cases.

Figure 153: Edge detection of the "Girl" image using a 5^2x50x1 neural network

Figure 154: "Girl" image edges arbitrated between the Roberts and Canny produced edge maps using a 5^2x50x1 neural network

Figure 155: "Girl" image edges arbitrated between the Roberts and Deriche produced edges maps using a 3^2x15x1 neural network

Figure 156: "Girl" image arbitrated between the Prewitt and Canny produced edge maps using a 5^2x40x1 neural network

Figure 157: "Girl" image edges arbitrated between the Roberts and Prewitt produced edge maps using a 5^2x20x1 neural network

This image is a difficult image for any edge detector due to the low contrast between different areas in the image. As for the squares image, edges produced by the neural network edge detector are better defined, being thin and continuous. However a large number of the lines corresponding to the low contrast edges are missing. All the arbitration schemes mark them, however a larger number of false edges are produced. A clear difference between the arbitrated results and the edge detection techniques alone, is the small round edge in the left side of the image which is completely missed by the majority of the edge detection techniques alone. This detail was also missed by the Roberts operator in the enhanced image, however it can be seen to be present in the arbitrated example using that filter. Arbitrated maps tend to reflect characteristics of the methods from which they were arbitrated, producing intermediate versions of the edges.

From the images presented it can be seen that edge maps that result from the arbitration present more scattered points than the map produced by the neural network edge detector. These are due to a more complete detection being carried out. which include edges that correspond to features in the background. Although incompletely marked the results are better than for the neural network edge detector which misses them completely. A larger number of edges are present in the maps produced by the arbitrator. This suggests that using the arbitration strategy enhancement carried out by the edge enhancement filters produces a more sensitive operator than a neural network alone.

6.7 Comparison Result

The selection between the arbitration scheme proposed and the neural network for edge detection is difficult to perform, due to the different and conflicting requirements for edge detection and the different characteristics of the produced maps. Both techniques correctly located the edges and presented some degree of noise robustness. Edges produced by the neural network detector are thinner but incomplete (or less complete) than the arbitration cases investigated. The arbitration cases seem to require a stronger definition of the learning set to achieve the same degree of thinness as the neural network edge detector.

6.8 Chapter conclusion

In this chapter the development of a neural network edge arbitrator was presented. This was applied to the edge map arbitration between pairs of edge enhancement schemes. Solutions obtained were presented and evaluated, as in the previous chapter. Advantages and disadvantages of the method have been shown. relative to the edge detection algorithms and to the neural network edge detector developed in Chapter Five.

The next chapter presents an overview of the work done and conclusions from the work presented so far. It evaluates the achievement of the objectives. Finally it suggests further research to be carried out.

7 CONCLUSIONS

"Valeu a pena?
Tudo vale a pena
se alma não é pequena"[15]
(Fernando Pessoa, in Mensagem)

7.1 Surcease

In this thesis research into a novel edge detection strategy has been carried out. The technique is based on the arbitration between edge enhanced maps produced by diverse edge enhancement filters. It was hoped that the different edge maps, through arbitration, would complement each other and through the process of merging would produce an unique edge map. This was expected to inherit some of the suitable characteristics from each of the source maps. An artificial neural network was used to perform the arbitration task. The system development has been presented along with examples that show its advantages and disadvantages.

In Chapter One the problem of edge detection was introduced and its importance within digital image processing and computer vision applications highlighted. This chapter also presented the objectives of the work to be carried out and an overview of the structure of the Thesis.

---

[15] Was it worthwhile?
Everything is worthwhile
if the soul is not insignificant

It was also explained, in this chapter, that in a large number of image processing applications, edge enhancement filters are expected to process a large number of images which have similar characteristics. Effectively, as images are produced by the same acquisition system they should present similar distortions (e.g., noise due to the acquisition process). Also, as similar subjects are investigated (e.g. blood cells) these will present similar border characteristics. This allows us to define samples of different types of image to which the system will be applied, from which representative sets of edge data can be extracted and used to train the artificial neural network. This will then be used to perform the arbitration task.

In Chapter Two a discussion of the concept of an edge was presented. It was important to analyse the concept of an edge as it defines the edge model used and thus affects the different algorithms or approaches selected. The performance of an algorithm is dependent on the edge model used, and the relative performance should be assessed with this constraint in mind. A large number of algorithms have been described in the literature, and a review of edge enhancement and detection techniques was next to be presented. This review aimed to cover a wide range of approaches that have been proposed. A more detailed description of the most widely referred to approaches in the literature then follows. Their performance and practical use is assumed due to the number of citations. Thus their selection should form a suitable first choice of methods to be implemented and to be used for the testing of the arbitration strategy.

Comparisons of the edge detection algorithms was also carried out, as it could provide an additional criteria for the selection and grading of the algorithms.

However. comparing the published results on a like by like basis proved to be difficult and thus added little to the global grading of the algorithms presented.

Edge detection *figures of merit* were next to be considered. Figures of merit were proposed to perform a grading of the performance of edge detection algorithms. They incorporate some of the characteristics that are required to produce a good edge map. However, as they measure the overall performance, they are inappropriate for use in a development study as they blend the distinct aspects that should be taken into account.

A literature review of artificial neural networks has also been carried out and was presented within Chapter Two. Main paradigms were presented and discussed. so as to allow for the selection of the most suitable network to be implemented. Among several classic paradigms multi-layer perceptrons seemed to be the most suitable. as they do not require a complete definition of the learning sets and the large number of applications reported in the literature suggest their high potential for the problem in hand. This type of network was described in more detail later on as was the learning method used.

Mathematical proofs that show. or at least suggest. that the proposed arbitration scheme is feasible are presented in Chapter Three. These are followed by the detailed presentation of the multi-layer perceptron learning law used in the development of the arbitration technique.

Image processing applications. due to the large number of operations involved are computationally intensive. To try and alleviate the extra load introduced through the addition of the neural network arbitrator system a parallel implementation of the system. upon a transputer platform. is described in

Chapter Four. Parallel processing concepts were first described. Next to be described was the transputer platform used and the particular approach adopted for the implementation of the arbitration system. Implementations based on distinct paradigms were performed in order to assess and compare the performance exhibited. These were based on the pipeline and data parallelism paradigms. Different process allocations were investigated, in order to assess the efficiency of various implementations so as to be able to optimise the resource allocation. Reductions in computing time of up to 2.5, in comparison to a single processor implementation, were obtained using a transputer network of 7 processors, configured using a data parallelism approach.

Chapter Five presents two necessary preliminary tasks that were carried out. The first is the selection of edge enhancement or edge detection algorithms to be implemented. Such a task requires the selection and definition of suitable images that demonstrate the diverse performance exhibited by the different edge detection algorithms presented. A large number of image types were processed. From these a small subset were selected that were representative and provided meaningful characteristics for the subsequent analysis. Artificially generated images and 'standard' images, as commonly used for comparative studies in the literature, for edge detection were chosen. As published comparison criteria were found to be inappropriate for the work, more stringent and objective comparisons were researched and are presented in Chapter Five. These were used as a complement to the visual comparisons, being used as the first assessment factor. Through visual comparisons it was possible to grade less successful solutions and to note the increase in the quality of the system response being obtained. However, they become increasingly inefficient and difficult to perform as the quality of the produced

edge maps was increased. Measurable characteristics from the images being used were researched and suitable methods to quantify them defined.

Also within Chapter Five the implementation of the selected edge detection schemes and the analysis of their performance is presented. It was shown that the edge enhanced maps produced by different edge enhancement algorithms present different characteristics. It was also shown that different edge segments can be marked by different algorithms. The results are presented as processed pictures. to which the previously defined measurements were applied. These were presented together with a discussion on their relative merits.

The development of a neural network for edge detection is next to be described. This problem requires smaller networks and thus it is faster to test than the arbitration strategy, allowing an easier coarse definition for the arbitration strategy to be developed in a more expeditious way. Learning set characteristics were researched as were some neural network characteristics that are not known 'a priori'. Examples of networks trained with learning sets extracted from different images were presented and the produced edge maps characterised. Also the experiments that justified the selection of several options were described. It was shown that a neural network can be trained with the sample images chosen and is able to successfully process images of a similar or different nature. Their performance is dependent on the image from which the learning set was extracted. and from the quantity of information included in the learning set. Comparisons were drawn between the different responses. This process shares characteristics with the arbitration process described later.

The development started with the use of very reduced training sets, which were useful to initially test the system. However, the quality of the solutions produced were very poor, which clearly suggested the requirement for more extensive sets. These were successively enlarged until the meaningful part of the image was included. This increased the system learning times. However, the time necessary to process one image with an artificial neural network is independent from the data used to teach it, and thus, practical applications of any of the approaches is not bounded due to this fact.

Solutions achieved are able to detect 'main' edges, presenting a certain degree of robustness. Some of the solutions obtained also present a directionality of the response, which suggests that a more elaborate definition of the learning set could be required.

The edge maps obtained present thin and well defined edges, even when the neural networks had been taught with sets extracted from simplified images.

The implementation of the arbitration system is described in Chapter Six. This system generalises the edge detection systems described before. The arbitration system inherits characteristics from the neural network edge detector described before. The arbitration strategy was tested using several pairs of edge enhancement and detection algorithms and respective solutions were characterised and compared.

Chapter Six also presents the development and performance of the proposed technique. The proposed technique is firstly explained with emphasis on the

practical problems that arose and the options inherited from the neural network edge detection scheme previously presented.

It has been shown that the arbitration system can perform well, if suitable edge enhanced maps are used. The arbitration will select points (or features) from each of the maps and thus the edge maps must include different features to allow the arbitration system to work. It was shown that such a system was able to select relevant points from each of the edge maps. The arbitration system was applied to four different pairs of edge enhancement filters, with different degrees of success. The edge maps produced presented some characteristics which were unfavourable when compared to the neural network edge detector. However, other characteristics were favourable. One such favourable characteristic was the arbitration systems ability to detect edges within low contrast images.

Four tests. to the arbitration strategy were performed using different sets of edge enhancement algorithms. The first test to the arbitration strategy was performed using the Roberts and Canny edge enhancement operators. as they produce largely different edge maps. The first mainly marking high contrast edges and the second marking a wider number of edges. including low contrast and noise related features. The nature of the neural network arbitrator was analysed and verified to be more complex than a logical point by point merging of the maps arbitrated upon.

Comparisons between the produced maps and the maps arbitrated upon showed the former to be better in many aspects. The arbitrated maps are more complete and have a better definition than the ones produced by the Roberts operator. This operator. as is common for the gradient based operators. in order to mark low contrast edges. requires a low threshold value. As a

consequence, high contrast edges are marked thicker, which affects their localisation and definition. Although the arbitrated case did not present a one pixel thickness, as in the case of the edges marked by the Canny operator, the number of points present in the edge map is smaller and correspond to the high contrast edges. A clear reduction in the noise related features is visible.

A second test was performed by the substitution of the Canny edge detector with a quicker recursive algorithm, namely the Deriche algorithm, which also belongs to an optimal approach. This detector presents a higher distinction between high and low contrast edges, and thus it was expected that a clearer map would be obtained. The arbitrated maps produced were more complete than the ones produced by the Roberts operator and present high contrast edges. More lines were produced and the object shapes were clearer than in the previous case. Effectively a dislocation of the edges, due to the gaussian standard deviation value used in the Canny operator implementation, originates a small error in the edge localisation which is reflected in the arbitrated map as sporadic points besides the lines. These points, although they did not interfere with the definition of the edge, degrade the visual aspect of it.

Next a search for the low contrast and more blurred edges was carried out. This was achieved by changing the Roberts operator for another which had stronger marking capabilities. For this purpose the Prewitt edge detection operator was used. Prewitt's edge maps presented stronger edges and better marking capabilities than the Sobel one and presented more homogeneous lines. The edges are also thicker than those produced by the Roberts operator and this fact is reflected in the arbitrated maps obtained. However, this system produced a large number of scattered points in the background without any clear advantage in the marked edges when compared with the previous cases.

A fourth test was performed using similar approaches, namely the Roberts and the Prewitt operators. In this case similar edge maps are used. These having points marked in the same position, and forming similar shapes. Here the arbitrated maps present the best edge shapes. They also present a smaller number of scattered points which can not be directly attributed to an edge. A comparison with the maps arbitrated upon reveals that they present an intermediate situation between the two previous maps, which although revealing the success of the arbitration system suggests the need for further research work .

7.2 Further work

The work so far has only investigated the arbitration of 4 different pairs of edge enhancement filters. More then just two sets of inputs to the system should be investigated. The computational load will increase, however the extra information offered by the other edge maps could be significant. Leading on from this the selection of which edge filters to use. with respect to the image under study, is not trivial and a further arbitration system should be developed that automatically selects the appropriate edge filters to use.

Other combinations of inputs to the arbitration system are also possible such as the substitution of the maps by the image itself. thus forming a problem with mixed characteristics when compared with the ones described above.

The edges obtained from the arbitration system are broader than the edges marked by the neural network edge detector. This could be due to a more complex space being generated and suggests that a larger number of training samples is required. Further research should be carried out to allow for a

larger number of maps to be used. Effectively the use of only two methods leaves ambiguities when one point is only marked in one of the edge enhanced maps.

A comparison of the solutions obtained, in the four tested cases, does not originate the same grading for the different images studied. This suggests that different edge enhancement tuples will give a different preference for different image types. This suggests that different tuples should be tested as a more application directed criteria for stopping the learning phase. For instance monitoring of the quality of the edge map that a particular learning iteration produces, could be used as a criteria to stop the learning phase and possibly improve the quality of the obtained solution.

Thus the exciting work described within this thesis opens the way to further investigations, using arbitration, into the important topic of edge detection.

8 BIBLIOGRAPHY

[1] Abdou I ;*Quantitative Methods of Edge Detection*, Image Processing Institute, University of Southern California, USCIPI Report 830, 1973

[2] Abdou I E, Pratt W K; *Quantitative Design and Evaluation of Enhancement / Thresholding Edge Detectors*, Proceedings of IEEE, vol 67, No5, May 1979

[3] Adhami RR. Brewer VE: *Edge Detection Using A Ciclic Ring Of Integers Modulo 7*; 21st Annual Southeastern Symposium on System Theory, pages 145-147, Talahassee. 1989

[4] Amini, A; Tewfik, AH; *Comparison Of Parametric And Non-Parametric Edge Detection Algorithms*, SPIE Vol 1244 Image processing Algorithms and Techniques. pages 417-428. 1990

[5] Argyle; *Techniques For Edge Detection*. proceedings of IEEE(Letters). vol 59. pages 285-286. 1971

[6] Asano.A: Itoh.K; Chioka. YI: *The Nearest Neighbour Median Filter: Some Deterministic Properties And Implementations*. Pattern Recognition. vol 23. No10. pages 1057-1066. 1990

[7] Barlaud M. Gaidon T. Mathieu P. Feaureau J C: *Edge Detection Using Recursive Biorthogonal Wavelet Transform*. pages 2553-2566. ICASSP91. 1991

[8] Bernsen JAC; *An Objective And Subjective Evaluation Of Edge Detection Methods In Images.* ; Philips Journal of Resesarch, vol 46, Part 2-3, pages 57-94, 1991

[9] Berzins, V; *Accuracy Of Laplacian Edge Detectors.* Computer Graphics and Image Processing 27, 195-210, 1984

[10] Blicher, A P; *Edge Detection And Geometrical Methods In Computer Vision*, Rep STAN-CS-85-1041/AIM, Fev 1985

[11] Bolon P., Araji; *Symmetrical Recursive Median Flters Applcation To Noise Reduction*, 5Th European Signal Processing Coference (EUSIPCO90),vol 03, pages 813-816; Barcelona, Spain.

[12] Bovik AC, Munson DC; *Edge Detection Using Median Comparisons.* Comp Graph and Image Processing, vol 33. No3, pages 377-389, 1986

[13] Brazakovic D, Patton R, Wang R: *Rule Based Multitemplate Edge Detector*: Computer Graphics Models and Image Processing n3 pages 258-268,May 1991

[14] Bruce V, Green PR; *Visual Perception, Physiology, Psychology And Ecology.* Laurence Erlbaum Associates Publishers, 2nd edition. 1990

[15] Bryant B; *Evaluation Of Edge Operators Using Relative And Absolute Grading,* IEEE Conf on Pattern Recognition and Image Processing, pages 128-145. 1979

[16] Cahill P T. Knowles R J R: *Software Reliability And Algorithm Validation For Medical Imaging: Performance Of Common Edge Detection Methods In Nuclear Medicine*: Medical Physics 12. No 5. pages 575-580. Sep/Oct 1985

[17] Canny J: *A Computational Approach to Edge Detection.* IEEE Pattern Analysis and Machine Intelligence PAMI-8. No6. Nov 1986

[18] Canny J; *Finding Edges And Lines In Maps*; MIT Artificial Intelligence Report, Cambridge, Rep AI-TR-720, Juin 1983

[19] Castan S, Zhao J, Shen J;*Optimal Filters For Edge Detection Methods And Results*; Lecture notes in computer Science, 1990, vol 427, pages 13-17, 1990

[20] Carriero N, Gelernter D; *How To Write Parallel Programs A First Course* MIT Press 1990

[21] Chedi K, Liao Q; *Edge Detection Method Using Visual Perception*, SPIE vol 1961 Visual Information processing II, pages 28-33, 1993

[22] Chen J S, Medioni G, Detection, *Localization And Estimation Of Edges*: IEEE Pattern Analysis and Machine Intelligence PAMI 11 No 2 pages 191-198 Feb 1989

[23] Chen M H , Lee D, Pavlidis T; *Residual Analysis For Feature Detection.* IEEE Pattern Analysis and Machine Intelligence PAMI-13, No 1, Jan 91

[24] Chialkowski M, Basak S, Pfeifer S, Parkey R. Pehock R; *Anatomy Sensitive Optimisation Of Edge Detection Algorithms For Mr Images Of The Lower Spine*, Proc of SPIE. vol 1349 Application of Digital image Processing, pages 431-440. 1990

[25] Clark JJ; *Authenting Edges Produced By Zero-Crossings Algorithms*: IEEE Pattern Analysis And Machine Intelligence Pattern Analysis and Machine Intelligence PAMI-11, No1, pages 43, Jan 1989

[26] Cok R: *Parallel Programs For The Transputer*, Prentice Hall, 1991

[27] Coter M, Chermont JL: *Precis D'analise D'images*, Presses du CNRS 1989

[28] Daniell CE, Kemsley HD, Lincoln WP, Baraghimian GA; *Artificial Neural Networks For Automatic Target Recognition*, Optical Engineering, Vol 31 No 12 December 1992

[29] Davies LS; *A Survey Of Edge Detection Techniques*; Computer Graphics And Image Processing 4, p 248-270, 1975

[30] Davies ER; *Machine Vision: Theory, Algorithms, Praticalities* - Academic Press, 1990

[31] Dayhoff JE; Neural Networks An Introduction. Van Nostrand Reinhold 1990

[32] Delp EJ, Chu CH; *Detecting Edge Segments*, IEEE SMC-15, No 1, pages 145-153, Jan/Feb 1985

[33] Deriche R; *Fast Algorithms For Low-Level Vision*, IEEE Pattern Analysis and Machine Intelligence PAMI- No 12, vol 1, Jan 1990

[34] Deriche R; *Optimal Edge Detection Using Recursive Filtering*; 1st International Conference on Computer Vision, London England, pages 501-505, 1987

[35] Dimitriou G; Tsanakas P, Papakonstantinou G; *The Multitransputer Implementation Of A Hierarchical Edge Detection Algorithm*, Parallel and Distributed Computing in Engineering Systems. Elsevier Science Publishers BV, 1992

[36] Douglas SC. Meng TH-Y: *An Adaptative Edge Detection Algorithm Using A Modified Sigmoid-Lms Algorithm*, 23 Asilomar Conference on Signal Systems and Computers. Pacifc Grove, 1989

[37] Eberlein RB: *An Iterative Gradient Edge Detection Technique*. Computer Graphics and Image Procesing, number 5, pages 245-253. 1976

[38] Eberhart R C, Dobbins R W; *Neural Network PC Tools*, Academic Press 1990

[39] Fleck MM; *Some Defects In Finite-Difference Edge Finders*, IEEE Pattern Analysis And Machine Intelligence PAMI 14 No3,March 92

[40] Flynn MF; *Some Computer Organisations And Their Effectiveness*, IEEE Transactions Computer C21, pages 948-960, 1972

[41] Fram JJR, Deutsch E; *On The Quantitative Evaluation Of Edge Detection Schemes And Their Comparison With Human Performance*; IEEE Trans on Computers. vol c24, No6. Junho 1975

[42] Frei W, Chen CC; *Fast Boundary Detection:A Generalization And A New Algorithm*; IEEE TC, Vol C26, No10, OCT 1977

[43] French P; *The Light Fantastic Medical Show*, New Scientist, vol 145 No 1968. pages 25- 29. 11 March 1995

[44] Galletty J: *Occam 2*; Pittman Publishing 1990

[45] Geiger D. Poggio T; *An Optimal Scale For Edge Detection*: IJCAI 87 Proc of the Tenth International Joint Conferenceon Artificial Intelligence. 1987, Milan. Italia. pages 745-748. 1987

[46] Geman D: *Stochastic Model For Boundary Detection*. Image and Vision Computing. vol 5. No2. pages 61-65. 1987

[47] Gibson P: Cowan CFN: *On The Decision Regions Of Multilayer Perceptrons*: Proceedings of IEEE. vol 78. No10. October 1990

[48] Gokmen M; *A Comparison On Edge Detection Algorithms Based On Gaussian Filtering And Iteratively Refined Regularization*, in Communication, Control and Signal processing, ed E Arikan, Elsevier Science Publishers, 1990

[49] Gonzalez RC, Wintz P; *Digital Signal Processing*, Addison-Wesley Publishing Company 1987

[50] Gregson PH; *Angular Dispersion Of Edgel Orientation - The Basis For Profile-Insensitive Edge Orientation*; Inteligent Robots and Computer Vision to Algorithms and Techniques. vol 1607, pages 217-224

[51] Griffith A K; *Edge Detection In Simple Scenes Using A Priori Information*, IEEE Transactions on Computers. vol C22, No 4 April 1973

[52] Haifeng Y, Mäkelä H; *A Fast Edge Detection Method For Navigation*, Artificial Neural networks, Elsevier Science Publishers, pages 1253-1256, 1991

[53] Hall EL; *Computer Image Processing And Recognition*; Academic Press 1979

[54] Hammerston. D; *Neural Networks At Work*, IEEE Spectrum 30-6, pages 26-32. Juin 1993

[55] Hammerston. D; *Working With Neural Networks*. IEEE Spectrum 30-7. pages 46-53. July 1993

[56] Hanckok. ER: Haindi M: Kittler. J: *Multiresolution Edge Labelling Using Hierarchical Relaxation*: 11th IAPR Int Conference on Pattern Recognition. pages 140-144. 1992

[57] Haralick RM. Lee JS: *Context Dependent Edge Detection And Evaluation*. Pattern Recognition. Vol 23 No 12. .pages1-19. 1990

[58] Haralick RM, Watson L; *A Facet Model for Image Data*; Computer Graphics And Image Processing 15, pages 113-129, 1981

[59] Haralick RM, Shapiro LG; *Computer And Robot Vision*, vol I Addison-Wesley Publishing Company, 1992

[60] Haralick RM, Lee, JS; *Context Dependent Edge Detection*; 1988 Conf on Computer Vision and Pattern Recognition, Ann Arbor pages 223-228, 1988

[61] Haralick RM; *Digital Step Edges From Zero Crossings Of Second Directional Derivatives*; IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI 6 n1 January 1984

[62] Hecht-Nielsen R; *Kolmogorov's Mapping Neural Network Existence Theorem;* Int Conference on Neural Networks III, IEEE Press , pages 11-13, 1987

[63] Hecht-Nielsen R; *Theory Of The Backpropagation Neural Network*: Int Joint Conf on Neural Networks IEEE Press. pages 593-605. June 1989

[64] Hecht-Nielsen R; *Neurocomputing*; Addison Wesley Publishing Company 1990 (reprint with corrections Jan 91)

[65] Hertz L. Shafer W, On the use of morphological operator in a class of edge detectors. Computer Vision and Image Processing. pages 25-54. 1992

[66] Heyden FV: *Evaluation Of Edge Detection Algorithms*: 3rd International Conference On Image Processing And Its Applications. Conference Publication 307. IEE. 1989

[67] Higgins. WE. Hsu CM: *Edge Detection Using 2D Local Strutural Information*: ICASSP 91. 1991

[68] Hopfield J J, *Neurons With Graded Response Have Collective Computational Properties Like Those Of Two State Neurons*; Proceedings of the NAtional Academy of Sciences 81, pages 3088-3092, 1984

[69] Hopfield JJ, *Unlearning Has A Stabilizing Effect In Collective Memories*; Nature 304: 158-159, 1983

[70] Hopfield JJ, *Neural Networks And Physical Systems With Emergent Collective Computational Abilities*, Proceedings of Natural Academy of Sciences, No 79 pages 2559-2558. 1982

[71] Hopfield JJ; Tank. D W; *Computing With Neural Circuits; A Model*, Science vol 233. 1986

[72] Hueckel MH; *An Operator That Locates Edges In Digitized Pictures*;J ACM, vol 18. n1. pages 113-125, Jan 1971

[73] Hummel RA; *Feature Detection Using Basis Functions*, Computer Graphics and Image Processing. No 9, pages 40-55. 1979

[74] Imme M; *A New Method For Edge Detection*; 3rd International Conference On Image Processing And Its Applications. IEEE 1987

[75] Irie B. Miyake S: *Capabilities Of Three-Layered Perceptrons*. Int Conference on Neural Networks. New York, IEEE Press. pages 641-648. 1988

[76] Jeong H. Kim CI:*Adaptive Determination Of Filter Scales For Edge Detection*: IEEE Transactions on Pattern Analysis and Machine Inteligence. vol 14. No5. May 1992

[77] Jeong H. Kim CI. Woo W-T: *Determining Optimal Scales For Edge Detection Using Regularization*: Proc of the 1991 Conf on Robotics and Automation. Sacramento. California.pages 1596-1601. April 1991

[78] Jou, I-C; Tsai, Y-C; *Transputer Based Back-Propagation Neural Net Emulation System*; 1990 IEEE International Symposium on Circuits and Systems, pages 1879-1882; 1990

[79] Kadar I; Kurz L; *A Class Of Robust Edge Detectors Based On Latin Squares*; Pattern Recognition, pages 329-339, 1979

[80] Kakarakala R, Hero AD; *On Achievable Accuracy In Edge Detection*; ICASSP 91, 1991

[81] Kay S M, Lemay G J; *Edge Detection Using the Linear Model*; IEEE Trans on Accoustics, Speech and Signal Processing, vol ASSP-34, No5, October 1986

[82] Kelly MD; *Edge Detection In Pictures By Computer Using Planning*, Machine Intelligence, B.Meltzer and D michier(eds), vol 6, pae 397-409, edinburgh University Press, 1971

[83] Kerr DA; Bezdek JC; *Edge Detection Using A Neural Network*; 1st Conf on Science of Artificial Neural Networks, SPIE, Orlando Florida, pages 510-521, 1992

[84] Kirsch R; *Computer Determination Of The Constituent Structure Of Biomedical Images*; Computers and Biomedical Research, vol 4, n3, pages 315-328, 1971

[85] Kitchen LJ, Malin JA; *The Effect Of Spatial Discretization On The Magnitude And Direction Response Of Simple Differential Edge Detection Operators On A Step Edge*; CVGIP 47, pages 243-258, 1989

[86] Kitchen L, Rosenfeld A, *Edge Detection Using Local Edge Coherence*, IEEE TSMC 11, No9, pages 597-605; 1981

[87] Klaus B, Horn P; *Robot Vision*, MIT Press, McGraw-Hill Book Company 1986

[88] Kohonen T; *Correlation Matrix Memories*, IEEE Transactions on computers, vol C-21, no4 Apr 1972

[89] Kohonen T; *The Self Organiziong Map*; Proceedings of IEEE, vol 78, no9, Sep 1990

[90] Koiran P; *On The Complexity Of Approximating Mappings Using Feedforward Networks*, Neural Networks, vol 6, pages 649-653; 1993

[91] Kozarev A, Topalovic M; *Quantitative Evaluation of Efficacy of the Tangential Operator for Edge Detection.*

[92] Kundu MK. Pal SK; *Edge Detection Based on Human Visual Response*; Int Journal of Systems Science. vol 19, No12 , pages 2523-2542. 1988

[93] Kundu A; *Robust Edge Detection*; Pattern Recognition. vol 23, n5. pages 423-440. 1990

[94] Kurková V: *Kolmogorov's Theorem and Multylayer Neural Networks*: Neural Networks. vol 5, pages 501-506. 1992

[95] Lacroix V. *Edge Detection: What About Rotation Invariance*. Pattern Recognition Letters n11. pages 797-802. Dec 1990

[96] Lee D; Detection. *Classification And Measurement Of Descontinuities*. SIAM Journal of Scientific and Statistical Computing. vol 12. Part 2 pages 311-341. 1991

[97] Lee JSJ. Haralick RM. Shapiro LG: *Morphologic Edge Detection*. IEEE Journal of Robotics and Automation. vol RA-3. no2 April 1987

[98] Lindsay PH. Norman DA: *Human Informatiom Processing, An Introduction to Psychology*, Harcourt Brace Jovanovich Publishers 1977

[99] Lippman RP; *An Introduction To Computing With Neural Nets*; IEEE AASP magazine pages 4-23, April 1987

[100] LU Y, Jain RC; *Resasoning About Edges In Scale Space*; IEEE Pattern Analisys and Machine Intelligence PAMI 14, No 4, 1992

[101] Lu Y, Jain RC; *Behaviour of Edges in Scale Space*; IEEE Pattern Analisys and Machine Intelligence PAMI 11, No 4, 1989

[102] Lunscher WHHJ, Beddoes MP; *Optimal Edge Detector Design Part II: Coefficient Quantization*; IEEE Pattern Analysis and Machine Intelligence PAMI--8, No2, pages 178-187.March 86

[103] Lunscher WHHJ, Beddoes MP; *Optimal Edge Detector Design Part I: Parameters Selection and Noise Effects*; IEEE Pattern Analysis and Machine Intelligence PAMI- -8, No2. pages 164-177.March 1986

[104] MacLeod, IDG. *Comments on "Techniques for Edge Detection"*, Proceedings of IEEE. vol 60. No 4, pages 344, 1972

[105] ManickamT, Misra P. *Application Of Recursive Filters In Edge Detection Of Images*, ICASSP 1991

[106] Marr D. Vision: *A Computational Investigation Into The Human Representation And Processing Of Visual Information*. W.H.Freeman and Co. San Francisco. 1982

[107] Marr D. Hildreth E; *Theory Of Edge Detection*: Proceedings of the Royal Society of London. ser B. No 207. pages 187-217. 1980

[108] Marr D; *Early Processing of Visual Information*: Philosphical Transactions of the Royal Society of London. Ser B, No 275. pages 483-524. 1976

[109] Martelli, A; *An Application Of Heuristics Search Methods To Edge Andd Contour Detection*; Communications of the ACM No 19, pages 73-83, 1976

[110] Martelli, A; *Edge Detection Using Heuristic Search Methods*, Comp Graphics and Image Processing, No1, pages 169-182, 1972

[111] May, D, '*The Transputer*' in Gordon Harp(eds), Transputer Apllications, Pittman Publishing, 1989

[112] Meer P, Wang S, Wechsler H; *Edge Detection by Associative Mapping*; Pattern Recognition vol22, No5. pages 491-503, 1989

[113] Mehrotra R, Namadur K R, Ranganatan N; *Gabor Filter-Based Edge Detection*; Pattern Recognition vol 25 n12 pages 1479-1494. 1992

[114] Merõ, Vassy; *A Simplified And Fast Version Of The Hueckel Operator For Finding Optimal Edges In Pictures*; Proc 4th International Joint Conference Artificial Intelligence. pages 650-655, 1975

[115] Micheli ED. Ottonelo P. Tore V; *Localization and Noise in Edge Detection*: IEEE Pattern ANALYSIS and Machine Intelligence. PAMI- 11. No 10, pages 1107-1117. Oct 1989

[116] Modestino. JW: Fries RW: Edge Detection in Noisy Images Using Recursive Digital Filtering; Computer Graphics And Image Processing No6. pages 409-433, 1977

[117] Montanari. U: *On The Optimum Detection Of Curves In Noisy Pictures.* Communication of ACM . 14. pages 335-345: 1971

[118] Morgenthaler DG; Rosenfeld A: *Multidimensional Edge Detection By Hypersurface Fitting*; IEEE Pattern Analysis and Machine Intelligence PAMI-3. No4. pages 483-486. July 81

[119] Müller, B; Reinhard, J; *Neural Networks, An Introduction*; Physics of Neural Networks, Springer Verlag, 1990

[120] Neuvo V, Hernonen P, Defee I; *Linear Median Hybrid Edge Detection*, IEEE Transactions Computer Systems CAS34, n11, pages 1337-1343, 1987

[121] Nevatia R; *Evaluation Of A Simplified Edge Detector*, Computer Graphics and Image Processing No 6, pages 582-588, 1977

[122] Nevatia R, Babu K; *Linear Feature Extraction And Description*, Computer Graphics and Image processing, vol 13, pag 257-269, 1980

[123] Nordstrom N; *Biased Ansidotropic Diffusion: A Unified Regularization And Diffusion Approach To Edge Detection*; Lecture Notes in Computer Science, vol 427, pages 13-17, 1990

[124] O'Gormann, F; *Edge Detection Using Walsh Functions*, Artificial Intelligence, No 10, pages 215-223, 1978

[125] Overington, I; Greenway, P; *Practical First Difference Edge Detection With Sub-Pixel Accuracy;* Image and Vision Computing, vol 5, No 3, pages 217-224, 1987

[126] Paik, J K; Braileau, J C; Katsggelos, A K; *An Edge Detection Algorithm Using Multistate Adalines*; Pattern Recognition, vol 25, n2, pages 1495-1504, 1992

[127] Paik, J K; Kataggelos, A K; *Edge Detection Using A Neural Network*; ICASSP90

[128] Patton, R; *Rule Based Gradient Edge Operator*; Proceedings of SPIE, pages 122-129, 1987

[129] Pavlidis T; *Strutural Pattern Recognition*; Springer Verlag, 1977

[130] Pearson D; *Image Processing*; McGrawHill 1991

[131] Peli T, Malah D; *A Study of Edge Detection Algorithms*; Computer Graphics And Image Processing No 20, pages 1-21, 1982

[132] Peng J, Rush P, Herrmann T; Levy PP; Elboustini; *Morphological Filters and Edge Detection Applications in Medical Imaging*; 1991 Annual Conference of IEEE Eng in Med and Biol Soc Orlando, Forida vol 13, pages 251-252, 1991

[133] Person, E; *A New Edge Detection Algorithm And Its Applications In Picture Processing*, Computer Graphics And Image Processing No5, pages 425-446, 1976

[134] Petrou M, Kittler J; *Optimal Edge Detectors For Ramp Edges*, IEEE Pattern Analysis and Machine Intelligence PAMI- 13, No5, pages 483-491, 1991

[135] *PIP user Manual*, Matrox, 1986

[136] Pitas; *Markovian Image Models For Image Labelling And Edge Detection*; Signal Processing, No 15, pages 365-374, 1988.

[137] Pratt WK; *Digital Image Processing*, Second Edition New York, Wiley InterScience. 1991

[138] Prewitt JMS: *Object Enhancement And Extraction*, in Picture Processing and Pychopictorics. Eds BS Lipkin and A Rosenfeld, Academic Press, pages 75-151, 1970

[139] Rimmer. S; *Bitmapped Graphics*, Windcrest McGraw-Hill 1990

[140] Roberts L; *Machine Perception Of Three Dimensional Solids*, 1965

[141] Robinson GS; *Edge Detection By Compass Gradient Masks*; Computer Graphics And Image Processing No6, pages 492-501, 1977

[142] Rosenfeld A, Thurston M; *Edge And Curve Detection For Visual Scene Analysis*, IEEE transactions on Computers, vl c-20, No 5, pages 562-569, 1971

[143] Rosenfeld A; *The Max Roberts Operator Is A Hueckel Type Detector*, IEEE Pattern Analysis and Machine Intelligence PAMI--3, No 1, pages 101-103, Jan 1981

[144] Rosenfeld A, Kak A; *Digital Image Processing*; Academic Press 1982

[145] Rumelhart DE, Hinton GE, Williams RJ; *Learning Internal Representations By Error Propagation* ; in Rumelhart DE, McClelland and the PDP Research Group (Eds), Parallel Distributed Processing, vol I- Foundations, MIT 1986

[146] Saito N, Cunnigham MA; *Generalized E-Filter And Its Application To Edge Detection*, IEEE Pattern Analisys and Machine Intelligence PAMI 12, n8, pages 814-817, 1990

[147] Sarkar S, Boyer KL; *On Optimal Infinite Impulse Response Edge Detection Filters*, IEEE Pattern Analysis and Machine Intelligence PAMI- 13,No 11, Nov 1991

[148] Schachter BJ, Rosenfeld A; *Some New Methods Of Detecting Step Edges In Digital Pictures*. Communications of ACM, vol 21, pages 172-176, 1978

[149] Serra J; *Image Analysis And Mathematical Morphology*, Academic Press, 1982

[150] Shanmugam KS, Dickey FM, Green JA; *An Optimal Frequency Domain Filter For Edge Detection In Digital Pictures*: IEEE Pattern Analysis and Machine Intelligence PAMI--1, No1, pages 37-49, Jan 79

[151] Shaw, GB; *Local And Regional Edge Detectors: Some Comparisons*, Computer Graphics And Image Processing No9, pages 135-149, 1979

[152] Shen J, Castan S; *An Optimal Linear Operator For Step Edge Detection*, CVGIP:GMIP, vol 54, No2, pages 112-133, March 1992

[153] Sher, D; *Derivation Of Edge Detection Algorithms From Sample Images*; Conference on Neural and Stochastic Methods for Image and Signal Processing, SPIE, pages 576-582, San Diego 1992

[154] Shou J, Shu P; *One-Pixel Wide Edge Detection*; Pattern Recognition vol 22, No 6, pages 665-673, 1989

[155] Sinchak P; *Theoretical And Experimental Study Of Edge Detection Techniques*; Soviet Journal of Remote Sensing, vol 9, No2, pages 344-352, 1991

[156] Sonka M, Hlavac V; Boyle R; *Image Processing And Machine Vision*, Chapman et Hall, 1993

[157] Sousa LAPS, Ferreira LFC, Piedade MS; *Algoritmos De Processamento De Imagem: Implementação E Comparação De Desempenho*; RecPad 89, pages II-25 a II-34, Porto 1989

[158] Spacek LA: *Edge Detection And Motion Detection*; Image and Vision Computing vol 4, n1 pages 43-56, 1986

[159] Spreeuwers. LJS: A Neural Network Edge Detector: SPIE vol 1451 Non Linear Image processing, pages 204-215. 1991

[160] Stevens. RT: *The C Graphics Handbook*. Academic Press 1992

[161] Sun Y; *Neural Network Approach For Classification Using Features Extracted By A Mapping*, Pattern Recognition Letters; No 14, pages 749-752. 1993

[162] Szeliski R; *Bayesian Modelling Of Uncertainty In Low Level Vision* Int journal Computer vision, vol 5, No 3, pages 271 - 301, 1990

[163] Tabbone S, Ziou D; *Efficient Edge Detection Using Two Scales*, 1993 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, pages 789-790, 1993

[164] Tagare HD, de Figueiredo RJP; *On The Localization Performance Measure And Optimal Edge Detection*, SPIE vol 1244 Image processing Algorithm and Techniques, pages 408-416, 1990

[165] Tan HL, Gelfaud SB, Delp EJ; *A Comparative Cost Function Approach To Edge Detection*, Transactions on Systems Man and Cybernetics TSMC vol 19, n6, pages 1337-1349, 1989

[166] Tan HL, Gelfaud SB, Delp EJ, *A Cost Minimization Approach To Edge Detection Using Simulated Annealing*, PAMI vol 18, n3, 1992

[167] Terry PJ, Vu D; *Edge-Detection Using Neural Networks*; Conference Record of the Twenty Seventh Asilomar Conf on Signals. Systems and Computers. pages 391-395, 1993

[168] Tewfik AH: *Singular Value Decomposition Applied To Edge Detection*. Proceedings of the SPIE. 1153. pages 214-224. 1989

[169] Thovey NK. Martinez MD; *A Comparison Of Different Intensity Gradient Formulae For Orientation Analysis Of Electron Micrographs Scanning*, vol 13. pages 289-298. 1991

[170] Torre V. Poggio TA: *On Edge Detection*; IEEE Pattern Analysis and Machine Intelligence PAMI--8. No 2. pages 147-163, March 86

[171] *Transtech TMB16 10 Slot PC Transputer Motherboard User Guide*, Preliminary, Reference TMB016MAN1090, 1990

[172] Ulupinar F, Medioni G; *Refining Edges Detected By A Log Operator* ; Proc of ICPR 88, Ann Arbor, pages 202-207, 1988

[173] Venkatesh S; Kitchen LJ; *Edge Evaluation Using Necessary Components*, CGMIP, No 1, pages 23-30, Jan 1992

[174] Vernon D; Machine Vision. *Automated Visual Inspection And Robot Vision*, Prentice Hall 1991

[175] Van Hulle MM, Tollenare T; *A Modular Artificial Neural Network For Texture Processing, Neural Networks*. Vol 6, pages 7-32. 1993

[176] Waard, WP; *Neural Techniques And Postal Code Detection*; Pattern Recognitio Letters, No 15, pages 199-205. 1994

[177] Wahl FM; *Digital Image Signal Processing*; Artec House, 1987

[178] Wang J; Mitra SK; *Edge Detection Based On Orientation Distribution Of Gradient Images*; ; ICASSP91

[179] Wasserman PD: *Advanced Methods In Neural Computing*, Van Nostrand Reinhold, 1993

[180] Wasserman PD: Neural Computing, Theory and Practice. Van Nostrand Reinhold, 1989

[181] Weschler H: Kidode M: *A New Edge Detection Technique And Its Implementation*. IEEE TSMC. 7 pages 827-836. 1977

[182] Wu L, Xie Z;*Scaling Theorems For Zero Crossings*, IEEE Pattern Analysis and Machine Intelligence PAMI--12, No 1, Jan 1990

[183] Yakimovsky; *Boundary And Object Detection In Real World Images;* J. ACM No 23, pages 599-618, 1976

[184] Yan H, Wu J; *Character And Line Extraction From Color Map Images Using A Multi-Layer Neural Network*; Pattern Recognition Letters. No 15, pages 97-103, 1994

[185] Yoo J, Delp EJ, Coyle EJ, Bouman CA; *Intensity Edge Detection With Stack Filters*; SPIE vol 1451 Nonlinear Image Processing II 1991

[186] Zerubia J, Chellappa R; *Mean Field Annealing For Edge Detection And Image Restoration*; Signal Processing V: Theories and Applications., L Torres, E Masgrau and MA Lagunas(eds), Elsevier, 1990

[187] Zerubia J, Chellappa R: *Mean Field Approximation Using Compound Gauss-Markov Random Field For Edge Detection And Image Restoration*: ICASSP 90, 1990

[188] Zhang TX; *An Adaptative Approach to Edge Detection Based on Visual Perception.* Proceedings of International Conference IEEE in Medicine and Biology Society, vol 005. 1928-1929. 1991

[189] Zhou JY, Wu LD; *An Ai Structural Approach To Edge Detection.* in Syntatic and Strutural Pattern Recognition. ed by G Ferrare et al. Springer Verlag 1988

[190] Zhou YT. Venkateswar V. Chelappa R: Edge Detection and Linear Feature Extraction Using a 2-D Random Field Model: IEEE Pattern Analysis and Machine Intelligence PAMI-11 No 1 Jan 1989

A METHODS RETAINED

A.1 Introduction

This Appendix describes the implementation of the edge detection techniques used. As the majority of the techniques are convolution operators, this operator is presented first, and is common to all implementations unless otherwise stated (e.g. Deriche). For each of the techniques, the selected options are stated and the implemented mathematical formulae presented.

A.2 Convolution operators

Most local edge detection techniques are based on a convolution between the image and a filter, followed by a decision process. This decision process is not considered here. In the convolution, (also designated as edge enhancement), the filter is applied to all points of the image. This filter is usually defined by a matrix. This matrix is usually a square matrix of odd dimensions. This allows for the filter to be centred on a known point. In image terms, the filter is applied as a sliding window, and thus is commonly referred to as a 'window'.

This window is applied to all points in the image, producing a second image as the result. Usually the application is performed sequentially from left to right, and top to bottom, or in increasing values of the coordinates. Figure A.1 shows the values and notation used throughout this thesis. As a consequence, the filter elements are referred to according to the same sequence, thus for a 3x3 filter:

(n-1,m-1)	(n,m-1)	(n+1,m-1)
(n-1,m)	(n,m)	(n+1,m)
(n-1,m+1)	(n,m+1)	(n+1,m+1)

Figure A-1: Notation used

In all algorithms, the resulting image is scanned and the grey level span readjusted to 256 levels. Due to this the normalisation coefficient sometimes associated with some of the masks presented are omitted .

In the case of the borders, some values outside the image are needed. It was decided to allow a margin in the border of the image so that the filter could work. For comparison purposes the borders must not be included in the results since they will introduce transient effects due the strategy used, and thus it will produce spurious results.

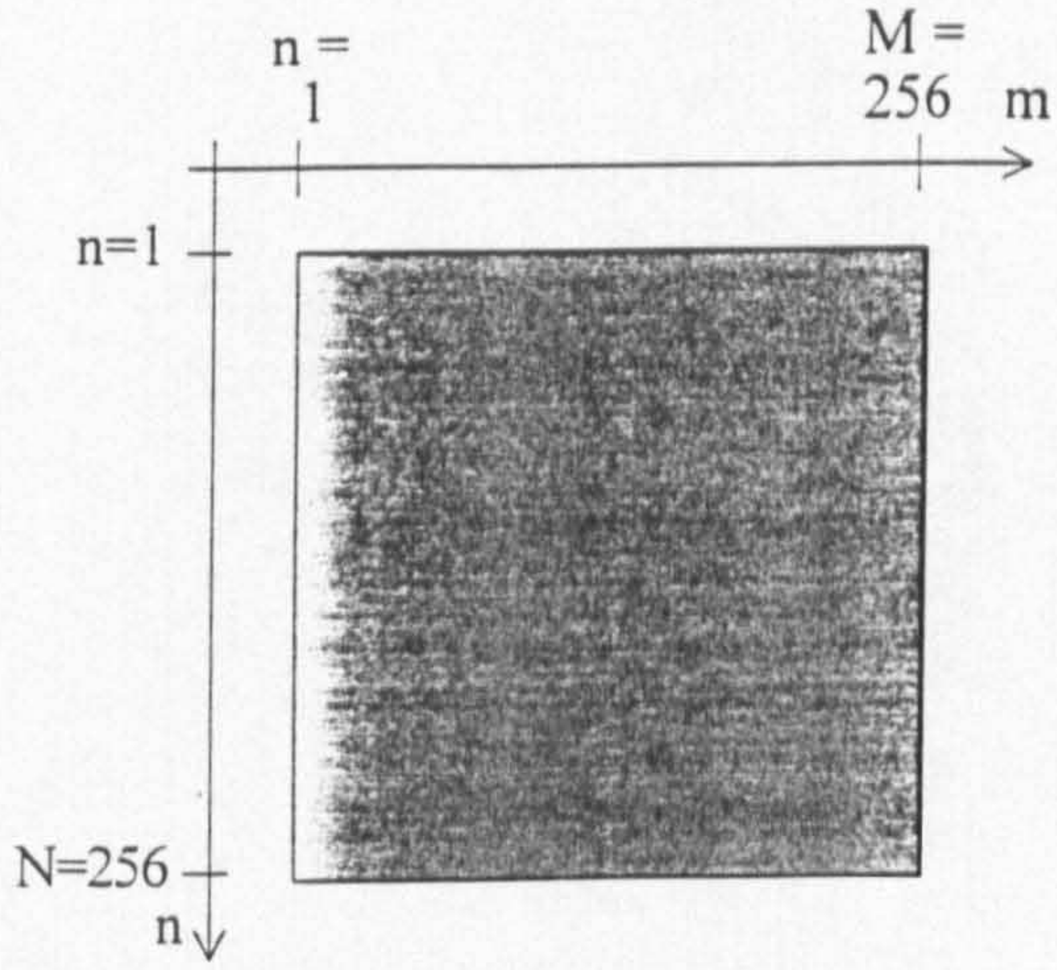

Figure A-2: Referential used

A.3 Implementation

All algorithms were implemented in Occam, working with a companion program in C. The Occam Toolset, and earlier TDS3, were used for producing bootable versions of the code. These receive and send the images to the companion C program, which retrieves, stores and displays the images.

A.4 Roberts Operator

The Roberts operator is simple and quick to use due to the small amount of operations involved. It is one of the few filters that use an even window. This characteristic implies an uncertainty in the marked position, to one pixel, in the filter. The principal disadvantage is also as a consequence of the small size of the window used, and is the noise sensibility. The result from the convolution filter requires the thresholding of the resulting image. Taking into account the isotropy, and, from Abdou results [1] , the square root model was

chosen. The Roberts gradient was implemented as

$$Rob = \sqrt{[p(n,m) - p(n-1,m-1)]^2 + [p(n-1,m) - p(n,m-1)]^2} \qquad \text{(A-1)}$$

which corresponds to the convolution with the filters

$$R_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \text{(A-2)}$$

and

$$R_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \qquad \text{(A-3)}$$

with

$$R_{x,y} = \sqrt{R_x^2 + R_y^2} \qquad \text{(A-4)}$$

The integer part of the square root $R_{x,y}$ is used. The image is scaled before returning to the computer.

A.5 Sobel

One of the most common edge detection algorithms is due to Sobel It is defined over a 3x3 window, and thus, is slightly slower to apply then the Roberts filter and slightly less sensitive to noise. It is also a convolution operator and requires thresholding of the edge enhanced map. Taking into account the isotropy the Square Root composition was chosen again. Sobel masks are defined as .

$$S_x = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \qquad S_y = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{A-5a,b}$$

and

$$S = \sqrt{S_x^2 + S_y^2} \tag{A-6}$$

The integer part of the square root is used and the resulting image is scaled as in the other cases so far.

A.6 Prewitt

Another 3x3 operator was proposed by Prewitt. It is also a convolution operator requiring the thresholding of the resulting edge map. Taking into account the isotropy the Square Root version was chosen again. Prewitt masks are defined as:

$$P_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix} \qquad P_y = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

and (A-7)

$$P = \sqrt{P_x^2 + P_y^2} \tag{A-8}$$

The integer part of the square root is used and the resulting image is scaled as in the cases so far.

A.7 Frei & Chen

Frei and Chen defined the edge detection problem as a multidimensional space and redefined a selection criteria based on the projection of the space defined by the following sets of vectors.

$$W_1 = \begin{bmatrix} 1 & \sqrt{2} & 1 \\ 0 & 0 & 0 \\ -1 & -\sqrt{2} & -1 \end{bmatrix} \qquad W_2 = \begin{bmatrix} 1 & 0 & -1 \\ \sqrt{2} & 0 & -\sqrt{2} \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} 0 & -1 & \sqrt{2} \\ 1 & 0 & -1 \\ -\sqrt{2} & 1 & 0 \end{bmatrix} \qquad W_4 = \begin{bmatrix} \sqrt{2} & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & -\sqrt{2} \end{bmatrix}$$

$$W_5 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & -1 \\ 0 & 1 & 0 \end{bmatrix} \qquad W_6 = \begin{bmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

$$W_7 = \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix} \qquad W_8 = \begin{bmatrix} -2 & 1 & -2 \\ 1 & 4 & 1 \\ -2 & 1 & -2 \end{bmatrix}$$

$$W_9 = \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad \text{(A-9)}$$

The decision process is defined by

$$F = \frac{\sum_{i=0}^{4} W_i}{\sum_{i=0}^{8} W_i} \qquad \text{(A-10)}$$

The value of F is considered for the edge map.

A.8 O'Gormann

The first algorithm of this type was developed by Huckel and presents several implementation problems. It uses a circular window and is based on computational heavy functions. A more untroublesome version was proposed by O'Gormann. This algorithm has the advantage of use mainly integers, which makes it better for implementation, particularly if T4 transputers are used.

The best fit of an ideal edge is performed over a window, being the decision based on the quality of the fit. The class of ideal edges considered are step edges of the form:

$$h(x,y) = \begin{cases} b+d & if\, x\cos\theta + y\sin\theta > 0 \\ b-d & otherwise \end{cases} \qquad \text{(A-11)}$$

O'Gormann uses the first six Walsh functions defined in a 4x4 window:

W_1=[1, 1, 1, 1. 1. 1, 1, 1, 1. 1, 1, 1, 1. 1, 1, 1]
W_2=[1, 1, 1, 1. 1. 1, 1, 1,-1.-1,-1.-1.-1.-1.-1.-1]
W_3=[-1,-1, 1. 1.-1.-1. 1, 1.-1.-1, 1, 1,-1.-1, 1, 1]
W_4=[-1,-1,-1.-1. 1. 1, 1, 1. 1, 1, 1, 1.-1.-1.-1,-1]
W_5=[-1,-1, 1. 1.-1.-1. 1. 1. 1. 1,-1.-1. 1. 1,-1.-1]
W_6=[-1. 1. 1.-1.-1. 1. 1.-1.-1. 1. 1.-1.-1. 1, 1,-1]

The coefficients for the template are

$$\begin{aligned} a_0 &= b \\ a_1 &= \tfrac{d}{2sc}[q(s+c) - q(s-c) - 2q(c)] \\ a_2 &= \tfrac{d}{2sc}[q(s+c) + q(s-c) - 2q(s)] \\ a_3 &= a_4 = a_5 = 0 \end{aligned} \tag{A-12}$$

where $s = \sin\theta$, $c = \cos\theta$ and

$$q(x) = \begin{cases} x^2 & if\, x \geq 0 \\ -x^2 & if\, x < 0 \end{cases} \tag{A-13}$$

The analytic distance between the points in the window and the ideal step is

$$\begin{aligned} b &= A_0 \\ d &= |A_1| + |A_2| \\ \tan\theta &= \begin{cases} d\,\mathrm{sign}\,\frac{A_1}{2A_2} & if \quad |A_1| \geq |A_2| \\ 2A_1\,\mathrm{sign}\,\frac{A_2}{d} & if \quad |A_1| < |A_2| \end{cases} \end{aligned} \tag{A-14}$$

where

$$A_i = \sum_{(x,y)\in R} p(x,y) f_i(x,y) \tag{A-15}$$

For the measure of the goodness of the match k is used. where k is:

$$k = \sqrt{\left[\frac{A_1^2 + A_2^2}{A_1^2 + A_2^2 + A_3^2 + A_4^2 + A_5^2}\right]} \tag{A-16}$$

For a perfect match k takes the value 1, and the operator rejects an edge if k<0.9. To deal with regions of uniform intensity the operator also rejects an edge if d<1.0

A.9 Laplacian

The Laplacian is a second order derivative operator, which, although having excellent prospects, is unacceptably sensitive to noise. As a second derivative it was expected to mark one pixel wide edges, if zero crossings were used. Thus no threshold or any other subsequent processing is required.

The Laplacian is computed using

$$\begin{matrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{matrix} \qquad \text{(A-17)}$$

As in the previous convolutions the resulting image is re-scaled.

A.10 LoG operator

As a solution to the above mentioned problem Marr and Hildreth proposed the use of the Laplacian of The Gaussian. The operator is the Laplacian applied to an image smoothed by a Gaussian function. This operation can be implemented as the convolution with the second derivative of the Gaussian function. This allows the implementation as a convolution operator with a very large mask, sometimes 32x32, to allow a variety of standard deviations for the Gaussian function. This operator also presents the characteristic of being separable, allowing, alternatively, to make separate convolutions in the vertical and horizontal directions.

A.11 Canny

Canny uses, the convolution in the edge direction with this function. The first derivative in the direction n is

$$G\bar{n} = \frac{\partial G}{\partial \bar{n}} = \bar{n} \bullet \nabla G \qquad \text{(A-18)}$$

n is ideally oriented normal to the edge direction. This direction could be estimated from the smoothed gradient direction by:

$$\bar{n} = \frac{\nabla (G * I)}{|\nabla (G * I)|} \qquad \text{(A-19)}$$

where * denotes convolution.

An edge point is defined to be a local maximum (in the direction of $\bar{n}$) of the operator $G\bar{n}$ applied to the image I. At a local maximum.

$$\frac{\partial}{\partial \bar{n}} G_{\bar{n}} * I = \frac{\partial^2}{\partial \bar{n}^2} G * I = 0 \qquad \text{(A-20)}$$

and the edge strength will be the magnitude of

$$|G_{\bar{n}} * I| = |\nabla (G * I)| \qquad \text{(A-21)}$$

Because of the associativity of the convolution. Canny first convolutes with a symmetric Gaussian G and then computes directional second derivative zeros to locate edges.

A more detailed description of Canny's algorithm is due to Fleck. which refers

to the original LISP code as reference. According to Fleck, Canny used the following masks (which are the 2x2 form of the Prewitt operator) to compute the X and Y components of the Gradient, which creates a systematic bias in the boundary locations:

$$\begin{bmatrix} -1 & 1 \\ -1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 \\ 1 & 1 \end{bmatrix} \tag{A-22}$$

Instead of these masks Fleck uses the masks [-1,0,1] to compute the first differences in four directions: horizontal H, vertical V and diagonals D1 and D2. This is a slightly different Sobel operator. The X and Y components of the Gradient are computed by projecting the diagonal differences onto the axes:

$$X = H + \frac{D_1 + D_2}{2} \tag{A-23}$$

$$Y = V + \frac{D_1 - D_2}{2} \tag{A-24}$$

and the amplitude is

$$A = \sqrt{X^2 + Y^2} \tag{A-25}$$

at each cell a value is interpolated for a pseudo-cell in the gradient direction. Values are extracted from the two neighbouring cells closest to the gradient direction: one from the H/V neighbours (Ahv) and one from a diagonal neighbour (A_{dd}). Let B be the larger of the magnitudes of X and Y, and S the smaller

The pseudo-cell's value is then

$$A_g = \frac{(B-S)A_h+SA_{ld}}{B} \tag{A-26}$$

A value A_{-g} for a pseudo-cell in the opposite direction is computed in the same way. A cell (x,y) is marked as a boundary if:

$$\begin{aligned} A(x,y) &\geq A_g(x,y) \\ A(x,y) &> A_{-g}(x,y) \end{aligned} \tag{A-27}$$

The value A is marked as the edge. A $\sigma = 4.0$ is implemented. This value was chosen as a compromise between the bias localisation minimisation and effectiveness for noise smoothing in the acquired pictures.

A.12 Deriche

The Deriche operator uses a smoothing function of the form $S = k(\alpha|n|+1)e^{-\alpha|n|}$ and leads to a recursive implementation being obtained using the following filter:

$$LL(m,n) = SS_{xx}(m,n) + SS_{yy}(m,n) \tag{A-28}$$

with the smoothing function

$$SS(m,n) = k(\alpha|m|+1)e^{-\alpha|m|}k(\alpha|n|+1)e^{-\alpha|n|} \tag{A-29}$$

with

$$SS_{xx} = k' m e^{-\alpha|m|} k(\alpha|n| + 1) e^{-\alpha|n|} \quad \text{(A-30)}$$

and

$$SS_{yy} = k(\alpha|m| + 1) e^{-\alpha|m|} k' n e^{-\alpha|n|} \quad \text{(A-31)}$$

with $k = \frac{(1-e^{-\alpha})^2}{1+2\alpha e^{-\alpha}-e^{-2\alpha}}$ and $k' = -\frac{(1-e^{-\alpha})^2}{e^{-\alpha}}$

Where $SS_{xx}(m,n)$ and $SS_{yy}(m,n)$ are the second directional derivatives in **x** and **y** respectively. The filter obtained is

$$LL(m,n) = e^{-\alpha|m|} e^{-\alpha|n|} - k\alpha|m| e^{-\alpha|m|} k\alpha|n| e^{-\alpha|n|} \quad \text{(A-32)}$$

and fixing k such that the output of the Laplacian to a constant input is null requires

$$k = \frac{1-e^{-2\alpha}}{2\alpha e^{-\alpha}} \quad \text{(A-33)}$$

To convolute a MxN input image x(m.n) with the filter. we separately perform the convolution with each term of the filter (A-32).

For the first term:

$$y_1(m,n) = x(m,n) + e^{-\alpha} y_1(m,n-1) \qquad \text{(A-34)}$$

for n=1,..,N and m=1,..,M with the boundary conditions x(m,0)=0 and $y_1(m,0)=y_1(m,-1)=0$ for m=1,..,M

$$Y_2(m,n) = e^{-\alpha} X(m,n+1) + e^{-\alpha} Y_2(m,n+1) = e^{-\alpha}[X(m,n+1) + Y_2(m,n+1)] \qquad \text{(A-35)}$$

for n=N,..,1 and m=1,..,M with the boundary conditions

$$x(m,N+1) = x(m,N+2) = y_2(m,N+1) = y_2(m,N+2) = 0 \qquad \text{(A-36)}$$

for m=1,...,M

and, a temporary image is obtained by adding the two vectors

$$r_1(m,N) = Y_1(m,n) + Y_2(m,n) \qquad \text{(A-37)}$$

for n=1....N and m=1....M

To r_1(m.n) is applied another filter in the vertical direction to obtain the result Y_a(m.n)

$$Y_1(m.n) = r_1(m,n) + e^{-\alpha} Y_1(m-1.n) \qquad \text{(A-38)}$$

for m=1....M and n=1....N with the following boundary conditions

$$r_1(0,n) = Y_1(0,n) = Y_1(-1,n) = 0 \qquad \text{(A-39)}$$

for n=1,..,N and

$$Y_2(m,n) = e^{-\alpha} r_1(m+1,n) + e^{-\alpha} Y_2(m+1,n) = e^{-\alpha} r_2(m+1,n) + Y_2(m+1,n) \qquad \text{(A-40)}$$

for m=M...,1 and n=1,..,N with the following boundary conditions

$$r_1(M+1,n) = r_1(M+2.n) = Y_2(M+1,n) = Y_2(M+2,n) = 0 \qquad \text{(A-41)}$$

and, the final result is

$$Y_a(m,n) = Y_1(m,n) + Y_2(m,n) \qquad \text{(A-42)}$$

for n=1,..,N and m=1.....M.

For the second term of LL(m.n)

$$Y_3(m.n) = x(m,n-1) + 2e^{-\alpha} Y_3(m.n-1) - e^{-2\alpha} Y_3(m,n-2) \qquad \text{(A-43)}$$

for n=1.....N and m=1.....M

with the boundary conditions x(m.0)=0 and Y_3(m.0)=Y_3(m.-1)=0 for m=1.....M

$$Y_4(m.n) = e^{-\alpha} x(m.n+1) + 2e^{-\alpha} Y_4(m,n+1) - e^{-2\alpha} Y_4(m.n-2) \qquad \text{(A-44)}$$

for n=N,..,1 and m=1,..,M with the following boundary conditions

$$x(m,N+1)=x(m,N+2)=Y_4(m,N+1)=Y_4(m,N+2)=0 \qquad \text{(A-45)}$$

for m=1,....,M

and, a temporary image is obtained by adding the two vectors

$$r_2=\frac{1-e^{-2\alpha}}{2}(Y_3(m,n)+Y_4(m,n)) \qquad \text{(A-46)}$$

To r_2(m.n) is applied another filter in the vertical direction to obtain the result y_b(m,n)

$$Y_3(m,n)=r_2(m-1,n)+2e^{-\alpha}Y_3(m-1,n)-e^{-2\alpha}Y_3(m-2,n) \qquad \text{(A-47)}$$

for m=1....M and n=1....N with the following boundary conditions

$$r_2(0,n)=Y_3(-1,n)=Y_3(0,n)=0 \quad for\ n=1,..,N \qquad \text{(A-48)}$$

and

$$Y_4=r_2(m+1,n)+2e^{-\alpha}Y_4(m+1,n)-e^{-2\alpha}Y_4(m+2,n) \qquad \text{(A-49)}$$

with the boundary conditions

$$r(M+1.n)=r(M+2.n)=Y_4(M+1.n)=Y_4(M+2.n)=0 \qquad \text{(A-50)}$$

for n=1....N

and, the final result is

$$Y_b(m,n) = \frac{1-e^{-2\alpha}}{2}(Y_3(m,n) + Y_4(m,n)) \qquad \text{(A-51)}$$

for n=1,...N and m=1,..,M

The Laplacian of the input image x(m,n) has to be computed by subtracting $y_b(m,n)$ from $y_a(m,n)$. It requires only 14 multiplication and 16 additions per output element independently of the size of the neighbourhood considered and specified by the parameter α.

This parameter specifies the size of the Gaussian, and is related to the standard deviation by

$$\sigma\alpha = \frac{\sqrt{}}{2\sqrt{\pi}} \approx \sqrt{2} \qquad \text{(A-52)}$$

APPENDIX
B

PUBLISHED PAPERS

Title	Conference
Integrating Parallel Edge Detection Schemes using Neural Network Arbitration	International Conference on Signal Processing Applications and Technology, ICSPAT '93 Pages 834-839, September 28 - October 1 Santa Clara, California, USA, 1993
Neural Network Arbitration of Edge Maps	World Transputer Congress, WTC'94 Pages 32-39, 5-7 September Cernobbio. Italia, 1994
Edge detection using Neural Networks : A comparison	International Conference on Digital Signal Processing, ICDSP'95 Pages 596-601. 26-28 June Limassol, Cyprus, 1995
Edge Detection using Neural Network Arbitration	5th International Conference on Image Processing and its Applications, IPA'95 Pages 514-518. 4-6 July 1995 Edinburgh. UK, 1995

INTEGRATING PARALLEL EDGE DETECTION SCHEMES USING NEURAL NETWORK ARBITRATION

MASN Ramalho ; KM Curtis
(1)Parallel Processing Specialist Group
Department of Electrical and Electronic Engineering
University of Nottingham
England

Abstract

Although it is not a recent research field, edge detection algorithms present some problems in many image processing applications. Different edge detectors present distinct and different responses to the same image, showing different detail. Our proposal is that different edge detection techniques be utilised in parallel, for generalised edge detection of different image types. This allows the attainment of several edge maps containing different features. From the various edge maps, edges are arbitrated with a neural network, performing the merging of the different maps. To overcome the increased computing time, implementation is being performed on a multi-transputer array, using the inherent parallelism of the techniques involved.

In this paper, the problem is presented along with the discussion of suitability for parallel processing. We present examples that show the different performance of the techniques for the same image. Results with a neural network are presented, along with a comparison with the individual techniques.

1. Introduction

Edge detection is an important problem in image processing applications as it is the common starting step for segmentation or feature detection in images. The idea of an edge is difficult to define precisely. In a picture a change in the colour/brightness of an object is perceived as a line that can be understood as an edge. Although not all changes in colour/brightness define a border of an object, it is generally accepted as a sharp change in the grey level intensity of a picture. Edges having various profiles are usually present in an image. From the definition the most typical profile is the step or roof edge, although it is rare in natural scenes. The edge profile usually appears distorted due to the image acquisition device and the noise that always appears in signals. Although in a natural scene the most common borders are vertical and horizontal lines, other shapes are also present and could be more significant in some applications. Many edge detection techniques have been developed to cope with this abundance of different types of edges.

2.1 A Review of Edge detection techniques

A wide range of approaches have been developed for solving the edge detection problem. Derivative methods are the most common. In this approach, the derivative of the picture is taken, and, using a threshold criteria, extreme values of the derivative are selected and accepted as edges. Usually this processing is done by the convolution of the picture with a mask [37, 38]. A different approach is template matching, where the window is matched with a set of templates of ideal edges. The template producing the highest correlation determines the existence of the edge at that point [13, 37, 39]. Although these masks are inherently discrete, some of the masks could define the sampling of a continuous bi-dimensional function [24, 25, 26]. Several methods could be used to produce families of masks [19]. A different approach, which avoids the threshold step is to use the zero crossing of the second derivative. Due to the frequency characteristics of the masks, some smoothing is applied, which causes the adverse effect of edge blurring. To alleviate this problem, factors can be included in the edge detector, e.g., the Laplacian of the image convolved with a Gaussian. Different support could be used to perform the smoothing and perform the derivatives, as a facet model [14]. An optimization of the shape of the smoothing function has also been performed [6]. The resulting function requires relatively large support, however a simplified version [6] and related versions [2,7,34,42] exist. Other operators have recently been proposed based on different filter support, as mathematical morphology [23,44], median filters [4,35] or others [22,41].

A different approach fits an edge model to a window of the image, and decides the presence/absence of the edge based on the quality of fitting [17]. The

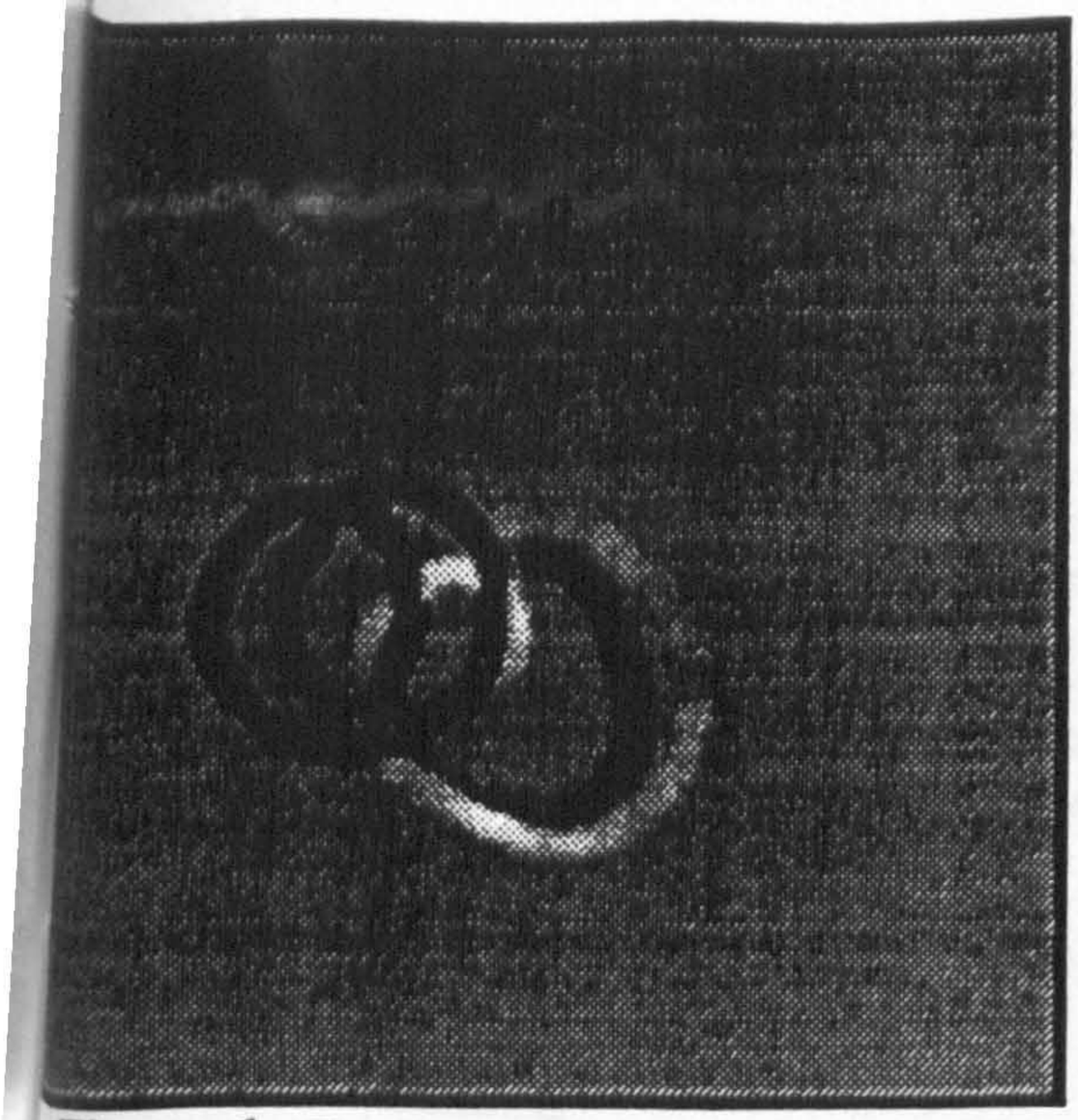

Figure 1 - Test Image. Random radius curve with random grey level

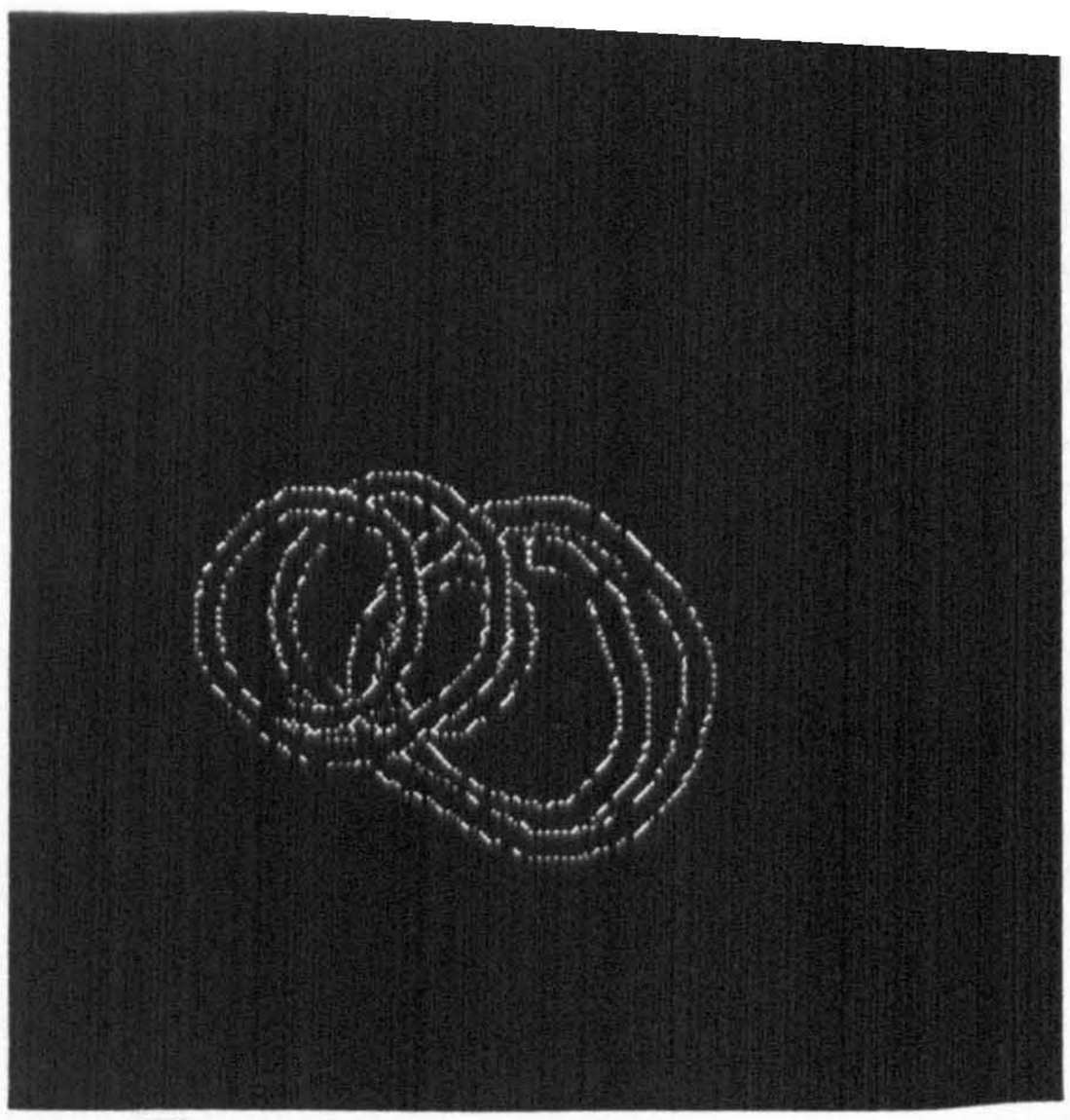

Figure 2 - Test Image - Reference Edges

algorithm is computationaly heavy and simplified approaches have been described [29, 31, 32]. Walsh functions have also been used, which, due to their binary characteristics allow a reduction of the computational complexity, and show a good ability to detect very low contrast edges. The technique assumes a rather slow ramp variation as a continuous edge. Statistical decision methods have also been applied [43]. The decision is based on the comparison of the characteristics on neighbouring windows. Other related approaches exist [5, 16, 20]. All the refired techniques operate in a local window. One of the main criticisms, is that the global context information of the picture is not considered. Some authors have tried several methods that use global information such has heuristic planning, graph search techniques or others [27, 28, 30]. Some of the above methods have been characterized and/or comparisons drawn between them.[3, 11, 12, 21, 36]. Despite all this effort a global comparison has not been achieved, due to several conflicting constraints that arise with different image processing applications.

2..2 The application of edge detection techniques

An efficient edge detector should achieve some desirable characteristics. Among these, resolution, precision, uniqueness, sensibility and robustness to image blurring and noise are important. In addition, the amount of operations involved should be as small as possible due to the usual size of images and eventual application constraints. Some of these characteristics are conflicting. The edges marked should be in the correct position, preferably thick and coherent and exempt of falsely marked edges. Unfortunately, most of the edge maps which are produced by edge detection algorithms do not fulfil these characteristics. Thus it is common to use some form of post-processing technique to perform single or isolated point removal, gap filling, etc.

Effectively edge detection using local algorithms consists mainly of convolving one or more masks, and thus is not only suitable, but highly suitable for parallel processing since all the operations, usually small in number, are independently repeated at all image points. We suggest that techniques be performed in parallel, with the edge detection results being arbitrated upon. Such a system mounted on a parallel processing platform would form a powerful general purpose edge detection tool as each technique is effective under different conditions. A neural network is an ideal means of carrying out the arbitration.

As can be seen from our discussions so far that many varied edge detection techniques exist. Each of these techniques have their advantages and disadvantages in the types of scenes which they are capable of analysing, their computational intensity and the ease of implementation on a processing platform. In most part of the cases the algorithms are ideally suited to being implemented on a parallel processing platform, if only through the use of data partitioning. From the above collection, we have selected several common operators representing different approaches.. The operators used are Roberts[38], Sobel[36] and

rewitt[37] from derivative approaches, Gormann[32] as a surface fit approach and Marr[26], Canny[6] and Deriche[9] operators as examples of optimal filters. Although artificial pictures do not present the difficulties that are present in natural scenes, the comparison of the methods is being done with generated pictures that try to reflect some of the features that are present in digitised video images, such as small blurring, for instance. This is done as we need to know the precise location of the edges., and thus allowing precise comparisons between the different techniques. An example of such a picture is shown in fig 1 , which consist of a random radius stripe, where the grey level changes

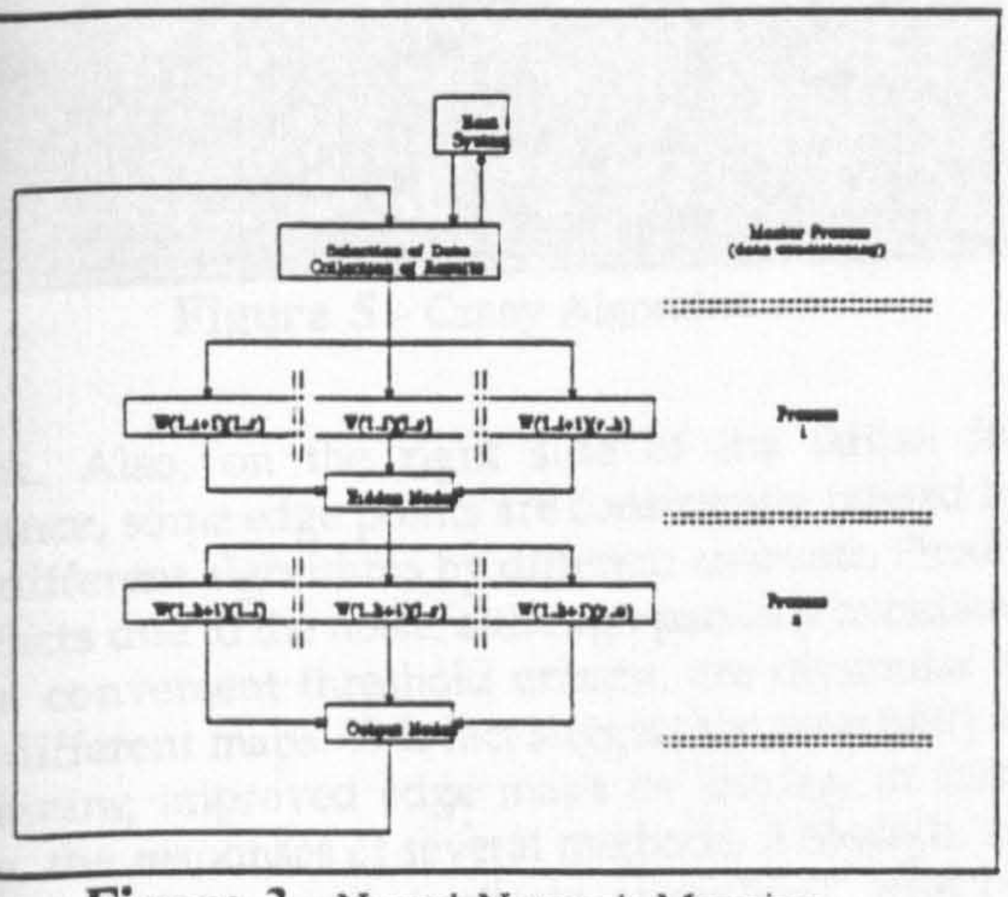

Figure 3 - Neural Network Mapping

with a random slope. To this picture noise or blurring could be added. In the presented case, the picture was blurred with a gaussian function, and a small amount of noise added. Several images are produced and tested. By the time of generation, a second image is produced representing the edge position in the image.

Artificial Neural Networks

Although not a recent field, Artificial Neural Networks have seen major development in recent years, and are being successfully applied to numerous problems. Among these, pattern recognition tasks are commonly cited in the literature.

Their capacity to handle incomplete or corrupted sets of data suggests that they can be applied to the referred recognition images and for instance, to infer the position of missing edges or misplaced edges based on the knowledge applied by the different edge detection techniques described.

The internal structure of a neural network allows the splitting of the network over several processors, with a relatively small amount of communication between them. Effectively, the architecture of a neural network naturally suggests a layer pipeline implementation, particularly relevant for intensive and repetitious applications such as edge detection This will reduce the overall processing time to the same order as that of a convolution filter of similar size. The only factors that must be propagated between different layers are the neuron vectors, which represent only a small fraction of the total amount of operations involved. For a nh1xnh2 coupling matrix there will be only n_{h1} values shared. Also, the massively parallel layer structure suggests the replication of the process over n processors.

4. Edge detection results

A representative number of the various edge detection schemes were implemented in the first instance on a single transputer, then on a multi transputer array. Results are presented in the form of processed images for different types of edge detector. Finally results are presented as to the effectiveness of a neural network.

Figures 4 through 6 show edgs maps obtained using several of the above mentioned methods. The pictures are presented, without further processing. It could clearly be seen that different methods detect different edge segments. The points falsely (?) marked on the interior of the band are not the same in the different

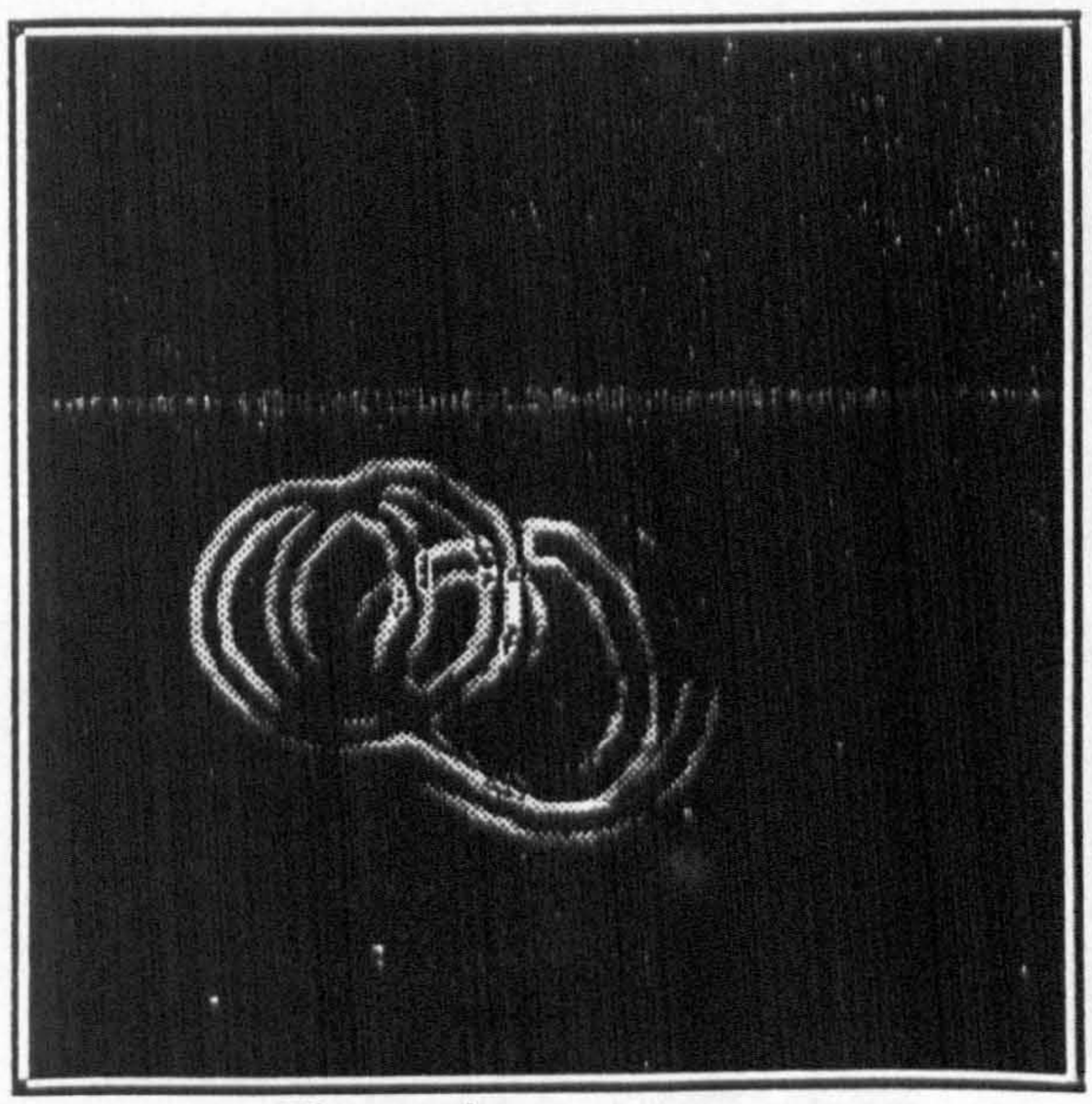

Figure 4 - Prewitt operator

Figure 5 - Canny Algorithm

maps. Also, on the right side of the stripe, for instance, some edge points are consistently missed by the different algorithms by different amounts. Finally artifacts due to the noise, although partially avoidable by a convenient threshold criteria, are dissimilar in the different maps. This fact suggest the possibility of obtaining improved edge maps by joining, in some way, the responses of several methods. Although, the processing time will duplicate or multipli, with the use of parallel processing techniques, it is expected to achieve an effective implementation of this process.

In the first instance we are researching the ability of a neural network for edge detection.. The obtention of a neural network of minimal configuration is important due to its intensive use. The learning set consists of images generated over a binary array by a small algorithm and representing edges at different orientations. A 5x5 window is used to supply the data to the input layer, followed by a 5 node hidden layer, and, in the picture, is plotted the result from the output node, without further processing. The examples shown, correspond to a learning set containing 32 elements. The learning set is generated in a square array, where different values. are assigned to the points above and below a reference line. Each element of the set corresponds to a different value of the slope . This procedure is allowed if we assume the constraint that edges in a small window are linear. This fact, that can not be assumed for larger windows, allows for the tests to be performed on the a automatic generation of edges.

A more natural and complete learning set could be extracted from the two picture-reference edges, although some problems arise. This learning set will present several problems due to the non equiprobability for the different pairs of pattern-responses. Effectively from the selection of the region of interest, it will be possible to perform the extraction of 41*55 elements, in which, 40 to 50% will represent the background, which, in a noise free image will be identical. In this case, a careful analysis of the learning set must be performed to avoid that large parts of the learning set represent the same pattern and to avoid inconsistency of the patterns in the learning set.

It is important to notice that, from a possible learning set consisting of 2^{25} points, a convergence to a working solution of the network was obtained with a small subset. Although the results are not "perfect", the simplicity and exiguity of the training set are good .

These problems are substantially aggravated for the arbitration system, since this requires that the learning set be extracted from several images, and thus the number of points will expand. Effectively, considering the fact that a neural net requires several hundred iterations to converge, the handling of such large amounts of data is a major problem for a efficient implementation of the learning phase.

5. Implementation

The system has been implemented on a transputer

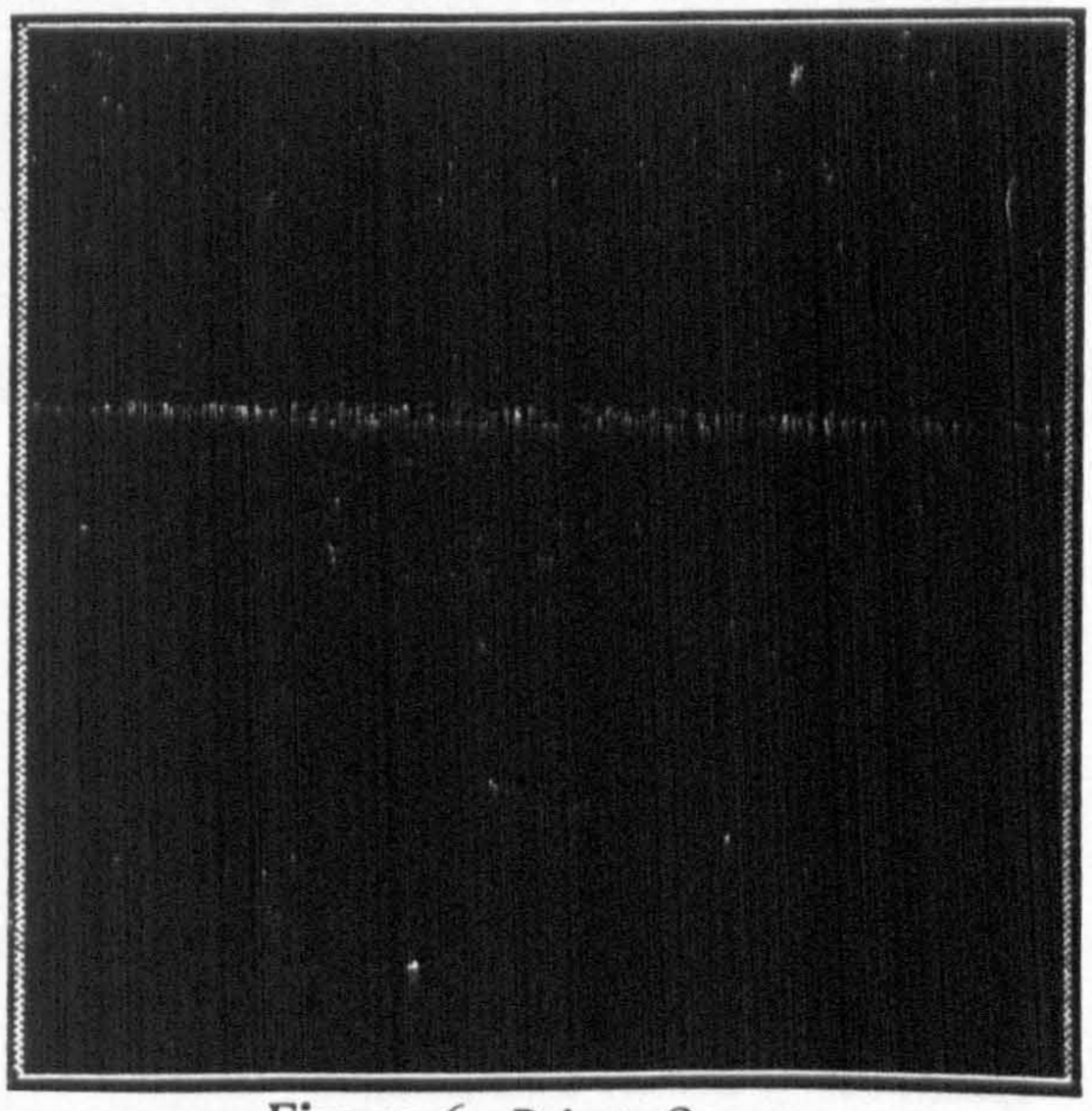

Figure 6 - Roberts Operator

ages are fed to the network by an external ogram, running in parallel on a host computer, hich accesses a Matrox frame grabber board. Edge etection algorithms are implemented using a data arallelism structure, splitting the image over n rocessors (Fig 4). The mapping of the neural network n the transputer array is done by layers. Each rocess includes a layer, so the extension of the layers n the neural network is done by adding intermediate rocesses. This allows a pipeline implementation of he network, overlapping process computing times. ach layer could also be mapped on a set of ransputers, which divide between them the coupling natrices, and share the node values as shown in igure 5.

Conclusion

We have shown that, distinct edge detection algorithm originate distinct edges maps. This maps presents some details that are particular to the output of each unique method. Also, the artifacts introduced by noise have diverse localization effects due to different frequency response of the methods. This suggest the fact that distinct algorithm responses could be integrated by an appropriate arbitration system. Although the complexity of the system is augmented, paralelization of the algorithms and an accurate division over an appropriate number of processors will minimize this problem. In the first instance the neural network has been tested for edge detection and found to be successful as shown in fig 7 and 8. Our investigations into the use of a neural network as an edge detector suggest that it is the ideal solution to the arbitration problem. The complete system is currently under development, and the initial results we have obtained look very promising.

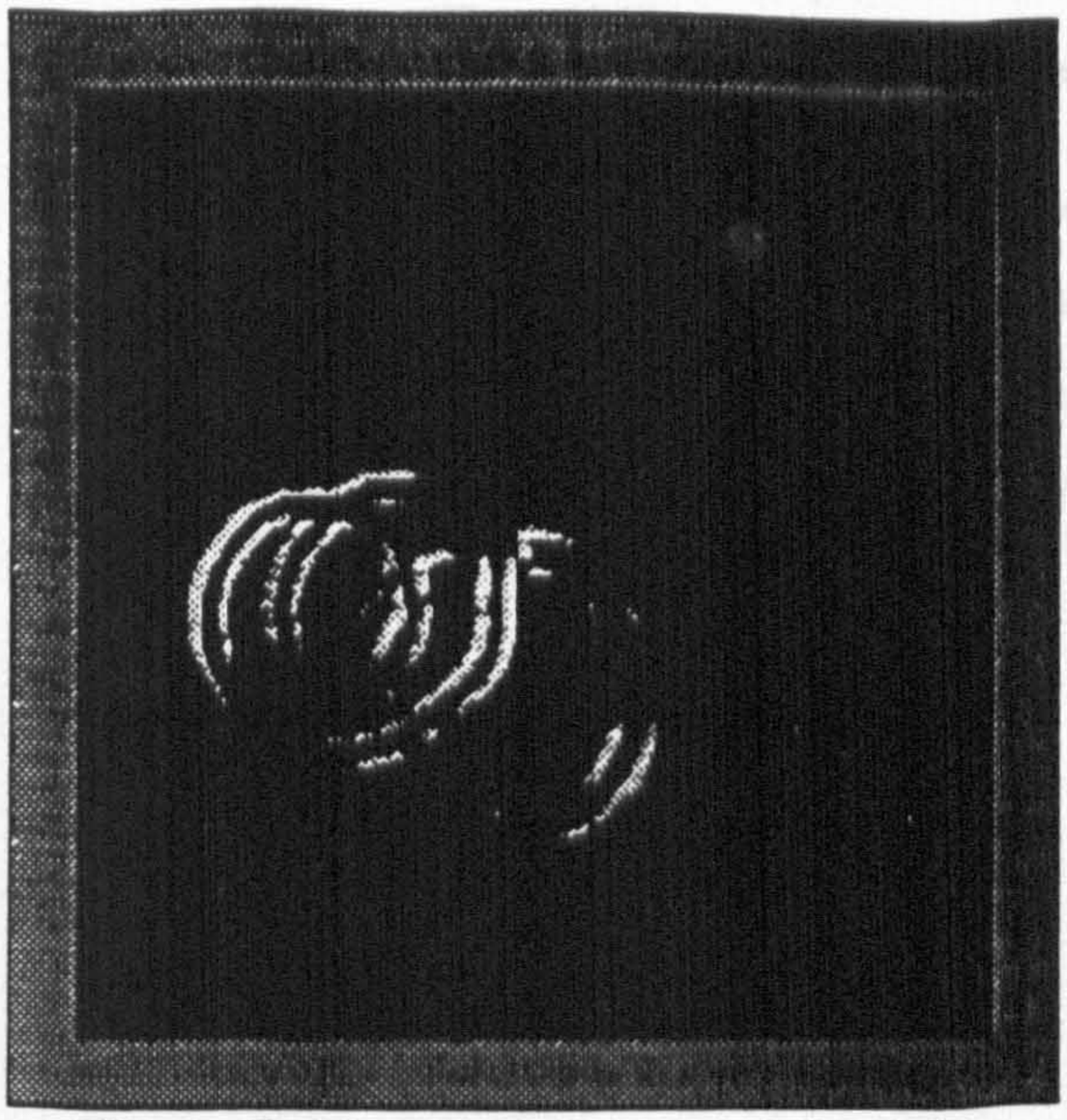

Figure 7 - Neural Network over a 7x7 window

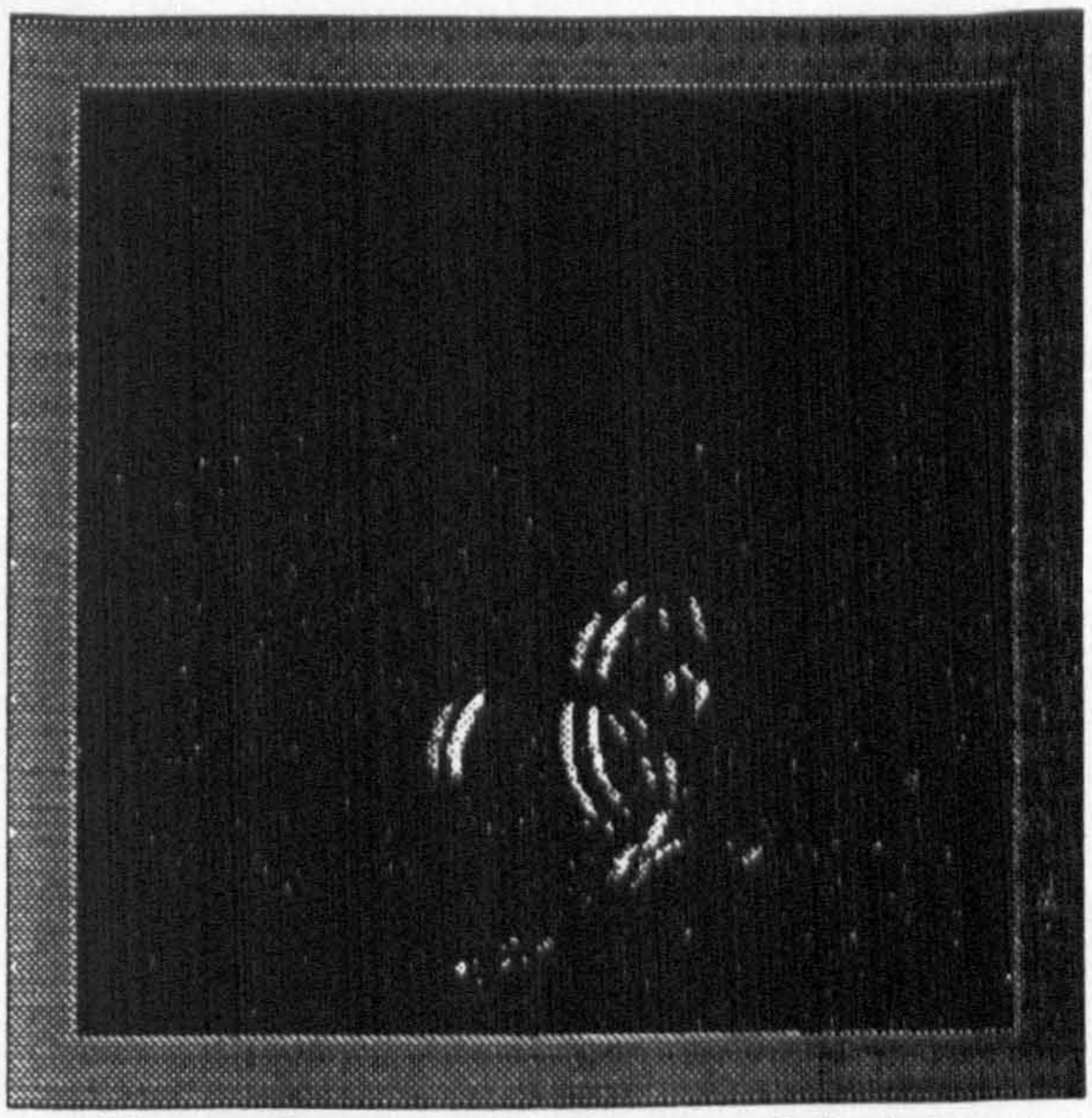

Figure 8 -Neural Network over a 3x3 window

Acknowledgments

MASN Ramalho would like to acknowledge the grant BD-1719 from Programa Ciência, JNICT, Portugal, under which his part of the work was carried out

References

[1] Abdou, IE; Pratt, WK - Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors, Proc of IEEE, vol 67, nº5, May 1979

[2] Barlaud, M; Gaidon, T; Mathieu, P; Feaureau, JC - Edge Detection using Recursive Biorthogonal Wavelet Transform, ICASSP 91, pp 2503-2566, 1991

[3] Bernsen, JAC - An Objective and Subjective Evaluation of Edge Detection Methods in Images. Philips Journal of Resesarch, vol 46, Part 2-3, pp 57-94, 1991

[4] Bolon, P;Raji, A; Lambert, P; Mouhoub, M - Symmetrical Recursive Median Filters Application to Noise Reduction: 5th Europpean Signal processing conf (Eusipco-90), Barcelona, Spain, vol 03, pp 813-816, 1990

[5] Bovik, AC; Munson, DC - Edge Detection Using Median Comparisons, Comp Graph and Image processing, vol 33, nº3, pp 377-389, 1986

[6] Canny, J - A Computational Approach to Edge Detection, J Canny, IEEE PAMI-8, nº6, Nov 1986

[7] Castan, S; Zhao, J; Shen, J - Optimal Filters for Edge Detection Methods and Results. Lecture Notes in Computer Science, 1990, vol 427, pp 13-17, 1990

[8] Clark, JJ - Authenting Edges Produced by Zero-crossings Algorithms: IEEE PAMI vol 11, nº1, pp 43, Jan 1989

Filtering; 1st International Conference on Computer Vision, London England, pp 501-505, 1987
[10] Deriche, R - Fast algorithms for low-level vision, IEEE PAMI nº 12, vol 1, Jan 1990
[11] Fleck, MM - Some Defects in Finite-Difference Edge Finders, IEEE Pami 14 nº3,March 92
[12] Fram, JJR; Deutsch, E - On the Quantitative Evaluation of Edge Detection Schemes and their Comparison with Human Performnce;IEEE Trans on Computers, vol c24, nº6, Junno 1975
[13] Frei,W; Chung-Ching Chen;Fast Boundary Detection:A generalizanon and a new Algorithm;IEEE TC, Vol C26, nº10, OCT 1977
[14] Haralick, RM - Digital Step Edges from Second Directional Derivatives, IEEE Pami 6, 1; Jan 1984
[15] Haralick, RM ; Shapiro, LG -" Computer and Robot Vision", vol 1, Addison-Wesley Publishing Co, NY, 1992
[16] Huang, T - Statistical Theory of Edge Detection; Comp Graphics and Image Processing, vol 43, pp 337-346, 1988
[17] Hueckel, M - An Operator That Locates Edges in Digitized Pictures;J ACM, vol 18, n 1, Jan 1971, pp 113-125
[18] Imme, M - A New method for edge Detection; 3rd International conference on Image processing and its applications, IEEE 1987
[19] Kadar, I; Kurz, I - A Class of Robust Edge Detection Methods Based on Latin Squares; Pattern Recognition, number 11, pp 329-379, 1979
[20] Kay, SM, Lemay, GJ - Edge Detection Using the Linear Model, IEEE Trans on Accoustics, Speech and Signal Processing, Vol ASSP-34, n 5 October 1986
[21] Kitchen, L; Rosenfeld, A - IEEE TSMC-11, n 9, pp 597-605, 1981
[22] Kundu, A - Robust Edge Detection, Pattern Recognition, vol. 23, nº 5, pp 423-440, 1990
[23] Lee, JSJ; Haralick, RM; Shapiro, LG - Morphologic Edge Detection, IEEE Journal of Robotics and Automation vol 3 , n 2, pp 142-156, 1987
[24] MacLeod, IDG - Comments on Techniques for Edge Detection, Proc of IEEE, vol 60, n 4, pp 394, March 72
[25] Marr, D - Early Processing of Visual Information, Philos Trans Royal Soc London, Ser B, nº 275, pp 483-524, 1976
[26] Marr, D; Hildreth, E - Theory of Edge Detection,, Proc Royal Soc of London, ser B, nº 207, pp 187-217, 1980
[27] Martelli, A - An Application of Heuristic Search Methods to Edge and Contour Detection, 1776
[28] Martelli, A - Edge Detection Using Heuristic Search Methods, Comp Graphics and Image Processing, nº1, pp 169-182, 1972
[29] Merö, Vassy - A Simplified and Fast Version of the Hueckel Operator for Finding Optimal Edges in Pictures, Proc 4th International Joint Conference Artificial Intelligence, pp 650-655, 1975
[30] Modestino, JW; Fries, RW - Edge Detection in Noisy Images Using Recursive Digital Filtering;; CGIP nº6, 1977, pp 409-433
[31] Nevatia- Evaluation of a Simplified Hueckel Edge-line Detector, CGIP nº6, pp 582-588, 1977
[32] O'Gorman, F - Edge Detection Using Walsh Functions, Artificial intelligence, nº 10,pp 215-223, 1978
[33] Pavlidis, T - Structural Pattern Recognition; Springer Verlag, 1977
[34] Petrou, M; KittlerJ - Optimal Edge Detectors for Ramp Edges, IEEE PAMI 13, nº5, pp 483-491, 1991
[35] Pitas - Markovian Image Models for Image Labelling and Edge Detection; Pitas, Signal processing, nº 15, 1988, pp 365-374.
[36] Pratt - Digital Image Processing, New York, Wiley interScience, 1978
[37] Prewitt, JMS; Object Enhancement and Extraction in Picture Processing and Psychopictorics, Eds BS Lipkin and A Rosenfeld, Academic press, pp 75-151, 1970
[38] Roberts, L - Machine Perception of Three Dimensional Solids,
[40] ; CGIP nº6, 1977, pp 492-501
Rosenfeld,A; Kak, A;- Digital Image Processing; Academic Press 1982
[41] Saito, N; Cunningham, MA; Generalized E-Filter and its Application to Edge Detection; IEEE PAMI, vol 12, number 8, pp 814-817, 1990
[42] Spacek, LA; Edge Detection and Motion Detection; Image and Vision Computing, vol 4, number 1, pp 43-56, 1986
[43] Yakimovsky - Boundary and Object Detection in Real World Images, J. ACM nº23, pp 599-618, 1976
[44] Yoo, J; Bouman; Delp, EJ; Coyle; EJ - Intensity Edge Detection With Stack Filters; SPIE vol 1451 Nonlinear Image Processing II 1991

Transputer Applications and Systems '94
A. De Gloria et al. (Eds.)
IOS Press

NEURAL NETWORK ARBITRATION OF EDGE MAPS

MASN Ramalho ; KM Curtis
Parallel Processing Specialist Group
Department of Electrical and Electronic Engineering
University of Nottingham
England

Abstract *Although it is not a new research field, current edge detection algorithms fail to offer a complete solution in many image processing applications. Different edge detectors present distinct and different responses to the same image, thus showing different detail. Our proposal is that different edge detection techniques be used in parallel, for generalised edge detection of different image types. This technique produces several edge maps containing different features. From the various edge maps, edges are then arbitrated with a neural network, which performs the merging of the different maps. To overcome the increased computing time, implementation is being performed on a multi-transputer array, using the inherent parallelism of the techniques involved.*

In this paper, the problem is presented along with a discussion as to its suitability for parallel processing. We present examples that show the different performances of the techniques for the same image. We also present the result of using a neural network to carry out arbitration of the image. A comparison is then drawn between the arbitrated image and the individual techniques.

1 -Introduction

Edge detection is an important role in image processing applications as it is the common starting step for segmentation or feature detection in images. The idea of an edge is difficult to define precisely. In a picture a change in the colour or brightness of an object is perceived as a line that can be understood as an edge. Although not all changes in colour or brightness define the border of an object, it is generally accepted that an edge is defined as a sharp change in the grey level intensity of a picture. Edges having various profiles are usually present in an image. From the definition the most typical profile is the step or roof edge, although it is rare in natural scenes. The edge profile usually appears distorted due to the image acquisition device and the noise that always appears in signals. Although in a natural scene the most common borders are vertical and horizontal lines, other shapes are also present and could be more significant in some applications. Many edge detection techniques have been developed to cope with this abundance of different types of edge.

2 -The Application Of Edge Detection Techniques

A wide range of approaches have been developed for solving the edge detection problem. Each of these techniques have their advantages and disadvantages in the types of scenes which they are capable of analysing, their computational intensity and the ease of

Figure 1 - Original

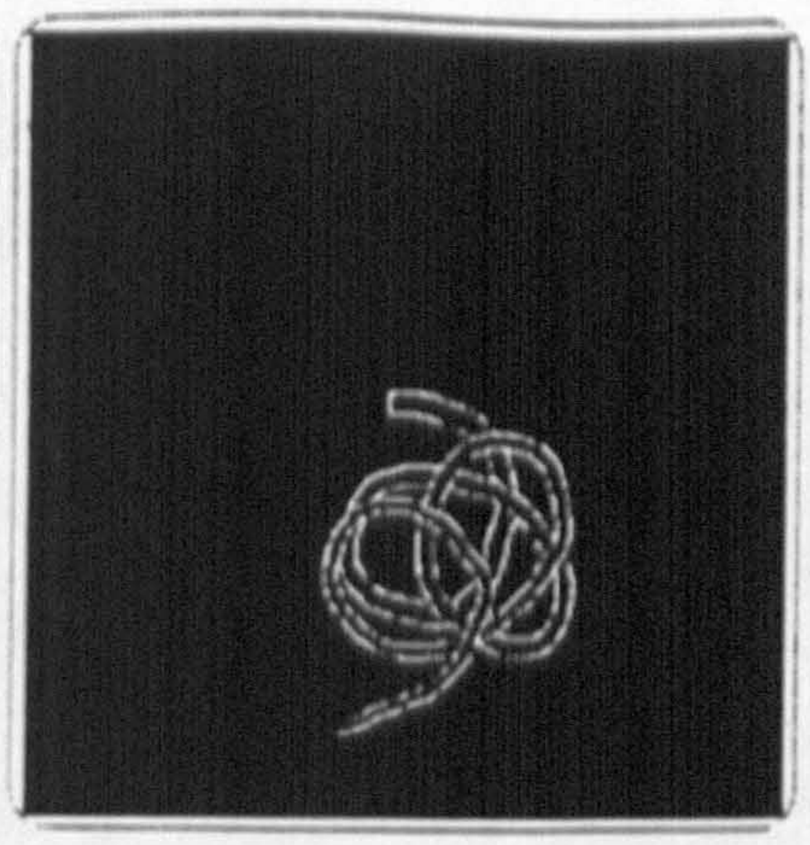

Figure 2 - Reference Edges

implementation on a processing platform. An efficient edge detector should achieve some desirable characteristics. Among these, resolution, precision, uniqueness, sensibility and robustness to image blurring and noise are important. In addition, the amount of operations involved should be as small as possible due to the usual size of images and eventual application constraints. Some of these characteristics are conflicting. The edges marked should be in the correct position, preferably thick and coherent and exempt of falsely marked edges. Unfortunately, most of the edge maps which are produced by current edge detection algorithms do not fulfil these characteristics. Thus it is common to use some form of post-processing technique to perform single or isolated point removal, gap filling, etc. Effectively edge detection using local algorithms consists mainly of convolving one or more masks, and thus is not only suitable, but highly suitable for parallel processing since all the operations, usually small in number, are independently repeated at all image points. We suggest that techniques be performed in parallel, with the edge detection results being arbitrated upon. Such a system mounted on a parallel processing platform forms a powerful general purpose edge detection tool, as each technique is effective under different conditions. The arbitration could be performed by the use of a rule based system. However, the wide range of methods and applications will require a multiple definition of the rule set used, according to the requirements of the different applications. Also, the large amount of data makes it difficult, if not, impossible, to define a general purpose tool. A neural network could overcome these difficulties and will be an ideal mean of carrying out the arbitration.

In most cases the algorithms are ideally suited to being implemented on a parallel processing platform, if only through the use of data partitioning. We have selected several common operators representing different approaches. In the case of derivative operators Roberts[8], Sobel[6] and Prewitt[7] were selected. The O'Gormann[5] operator was selected as a surface fit operator and Marr[4], Canny[1] and Deriche[2,3] operators were used as examples of optimal filters. Although artificial pictures do not present the difficulties that are present in natural scenes, the comparison of the methods has been carried out with generated images that try to reflect some of the features that are present in digitised video images, such as small blurring, etc. The artificially generated images were

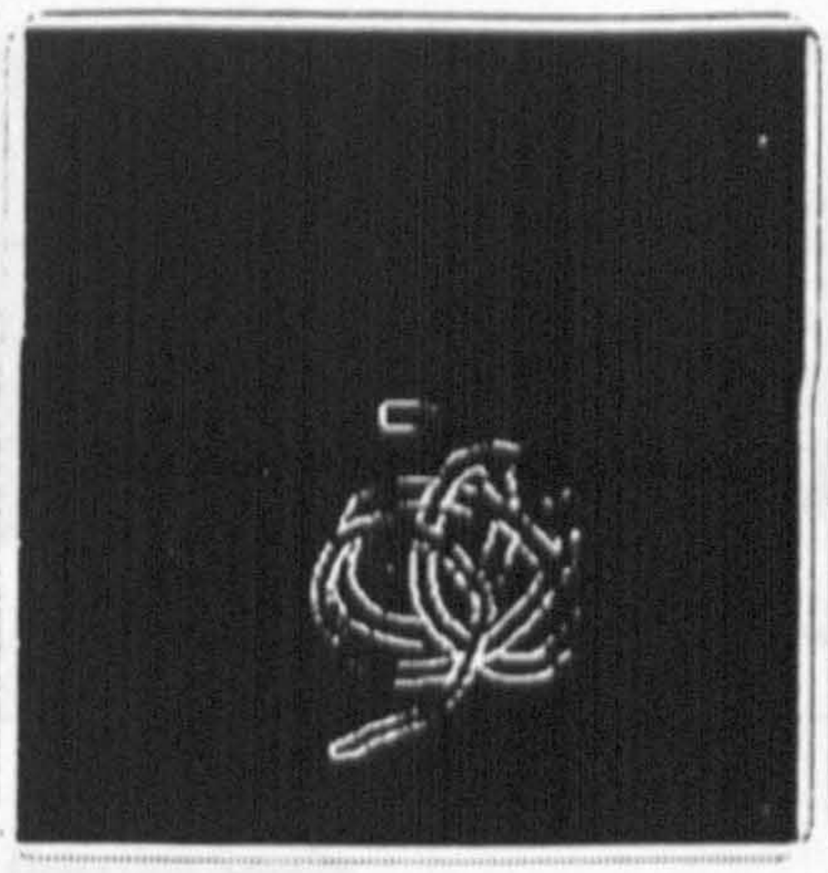

Figure 3 - Roberts Operator

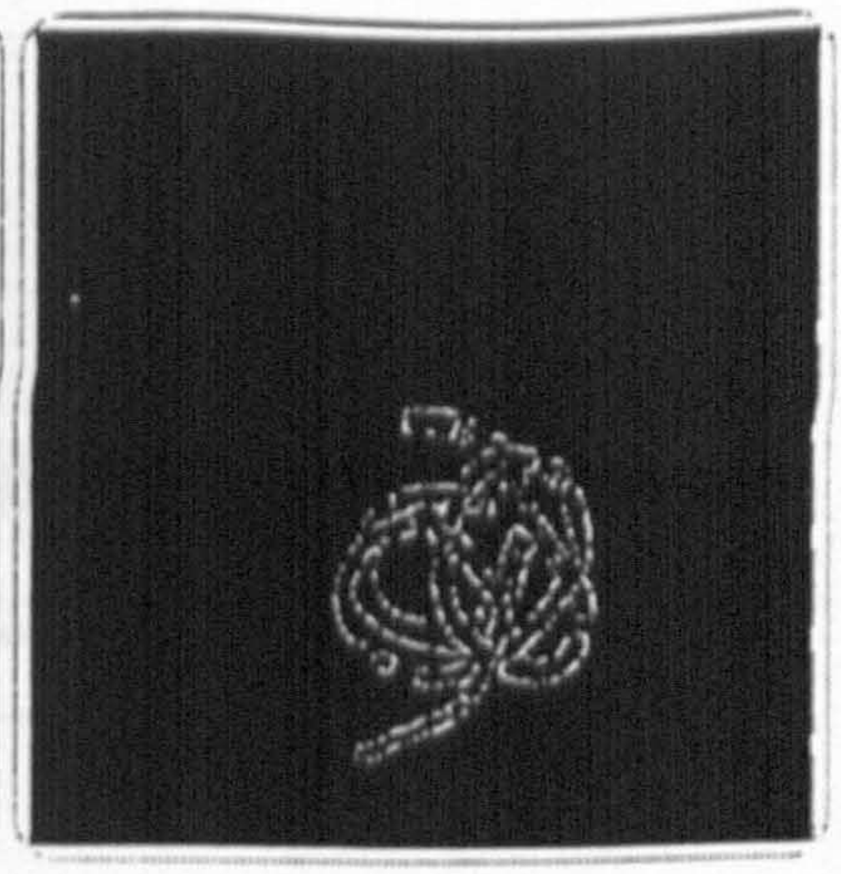

Figure 4 - Canny Operator

used as we need to know the precise location of the edges, thus allowing precise comparisons between the different techniques. An example of such a picture is shown in Figure 1 , which consist of a random radius stripe, where the grey levels change with a random slope. To this image noise or blurring could be added. In the presented case, the picture was blurred with a gaussian function, and a small amount of noise was added. Several images were produced and tested. At the time of generation, a second image (Figure 2) was also produced to show the edge position in the image.

3 -Artificial Neural Networks

Although not a recent field, Artificial Neural Networks have seen major development in recent years, and are being successfully applied to numerous problems. Among these, pattern recognition tasks are commonly cited in the literature. Their capacity to handle incomplete or corrupted sets of data suggests that they can be applied to the various edge images and for instance, to infer the position of missing edges or misplaced edges based on the knowledge supplied by the different edge detection techniques described. Among the different types of networks, multilayer perceptrons [9] are very common, and, this is the type of network used in this work.

The internal structure of this network allows the splitting of the network over several processors, with a relatively small amount of communication between them. Effectively, the architecture of a neural network naturally suggests a layer pipeline implementation, which is particularly relevant for intensive and repetitious applications such as edge detection This will reduce the overall processing time to the same order as that of a convolution filter of similar size. The only factors that must be propagated between different layers are the neuron vectors, which represent only a small fraction of the total amount of operations involved. For a nh1xnh2 coupling matrix there will be only nh1 values shared. Also, the massively parallel layer structure suggests the replication of the process over n processors.

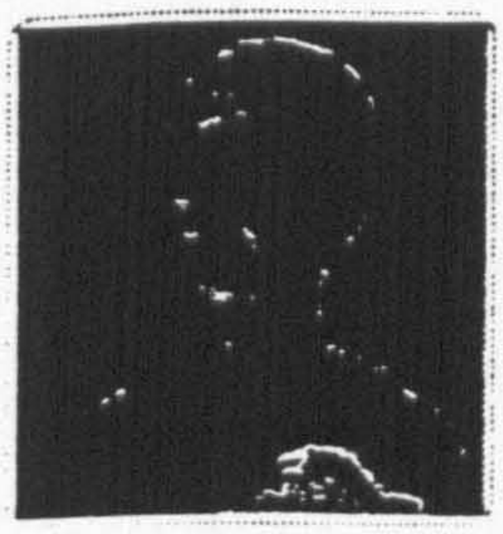

Figure 5 - Roberts Operator

Figure 6 - Original

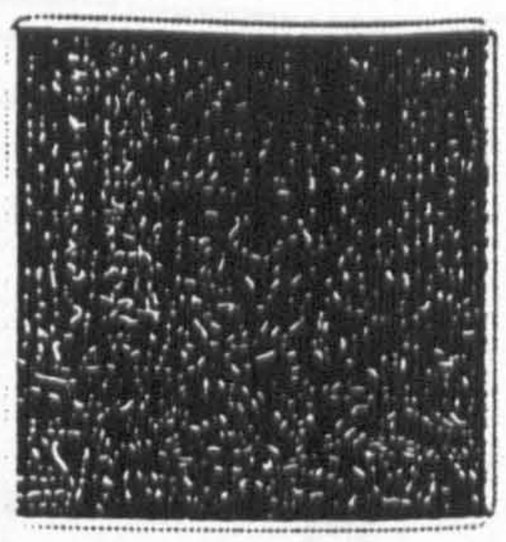

Figure 7 - Canny Operator

4 - Edge Detection Results

Several edge detection schemes were implemented on a multi transputer array. Results are presented in the form of processed images for different types of edge detector. Finally results are presented as to the effectiveness of a neural network arbitrator.

Figures 3, 4 and 5, 7 show edge maps obtained using two of the above mentioned methods. The pictures are presented without further processing. It can clearly be seen that the different methods detect different edge segments. The points falsely marked (?) are not the same in the different maps. Also, on the right side of the stripe, for instance, some edge points are consistently missed by the different algorithms by different amounts. Finally *artefacts* due to the noise, although partially avoidable by using a convenient threshold criteria, are dissimilar in the *different maps. This* fact suggests the possibility of obtaining improved edge maps by joining, in some way, the responses of several methods. Although the processing time will increase, with the use of parallel processing techniques, it is expected to achieve an effective implementation of this process.

A natural learning set could be extracted from the image and reference edges in pictures 1 and 2, although some problems arise. This learning set will present several

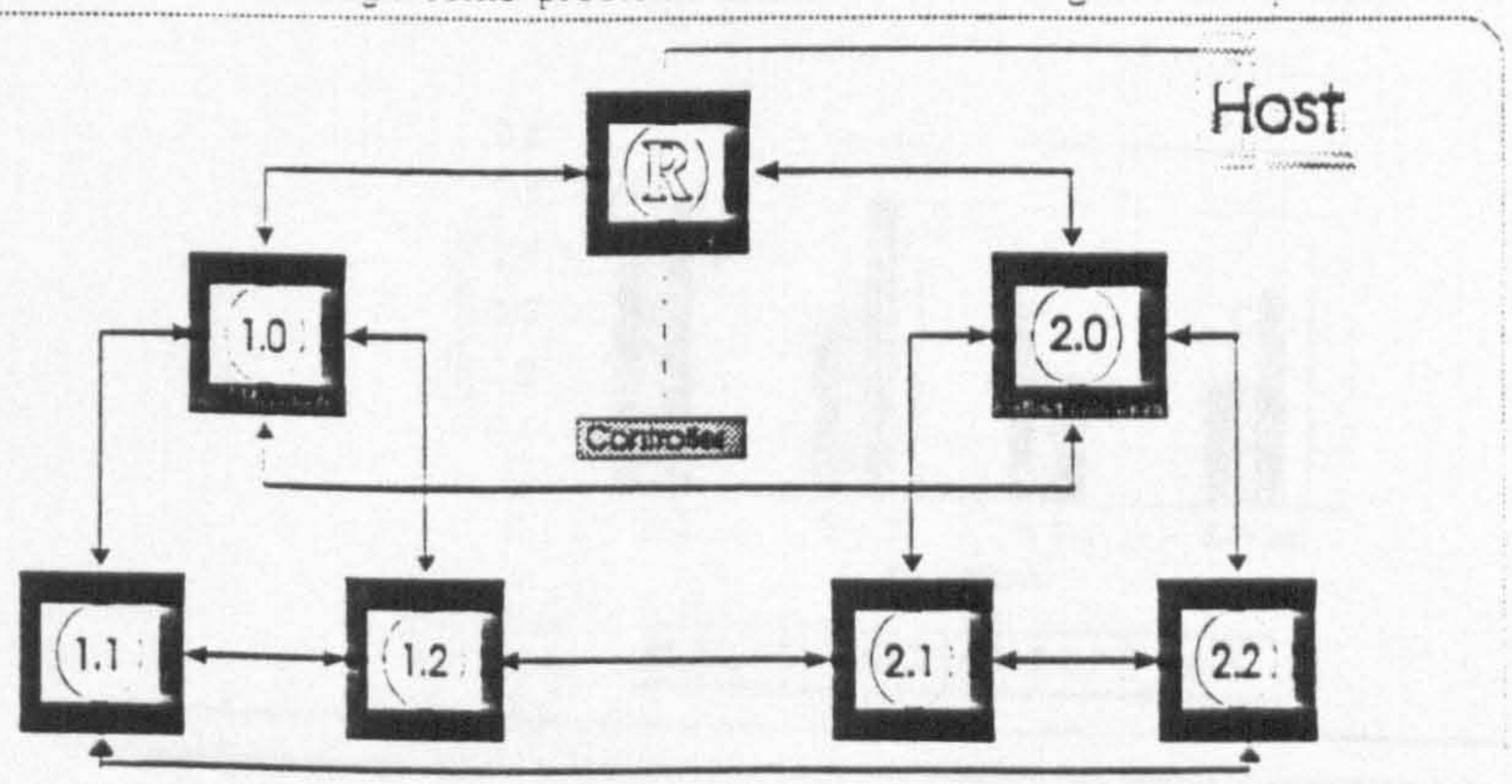

Figure 8 - Transputer Structure

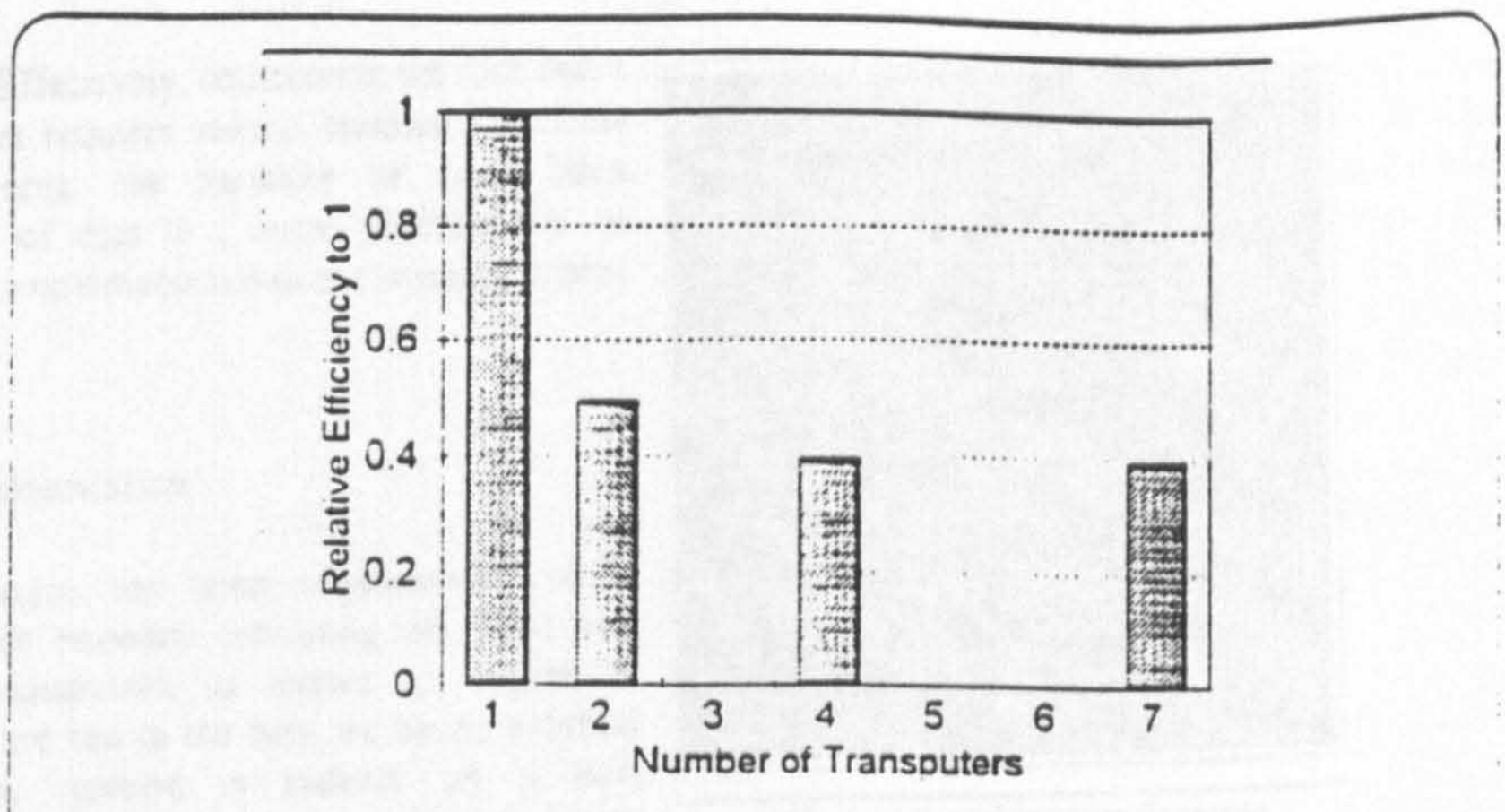

Figure 9 - Efficency obtained according to the transputer allocation

problems due to the non equiprobability for the different pairs of pattern-responses. Effectively from the selection of the region of interest, it is possible to perform the extraction of 84*110 elements, in which, 40% represent the background, which in a noise free image will be identical. In this case, a careful analysis of the learning set must be performed so as to avoid the large parts of the learning set which represent the same pattern and to avoid inconsistency of the patterns in the learning set. It is important to notice that, from a possible learning set consisting of 225 points, convergence to a working solution of the network can be obtained with a small subset. Although the results are not "perfect", the capabilities of such a training set are still under consideration.

These problems are substantially aggravated for the arbitration system, since this requires that the learning set be extracted from several images, and thus the number of points will

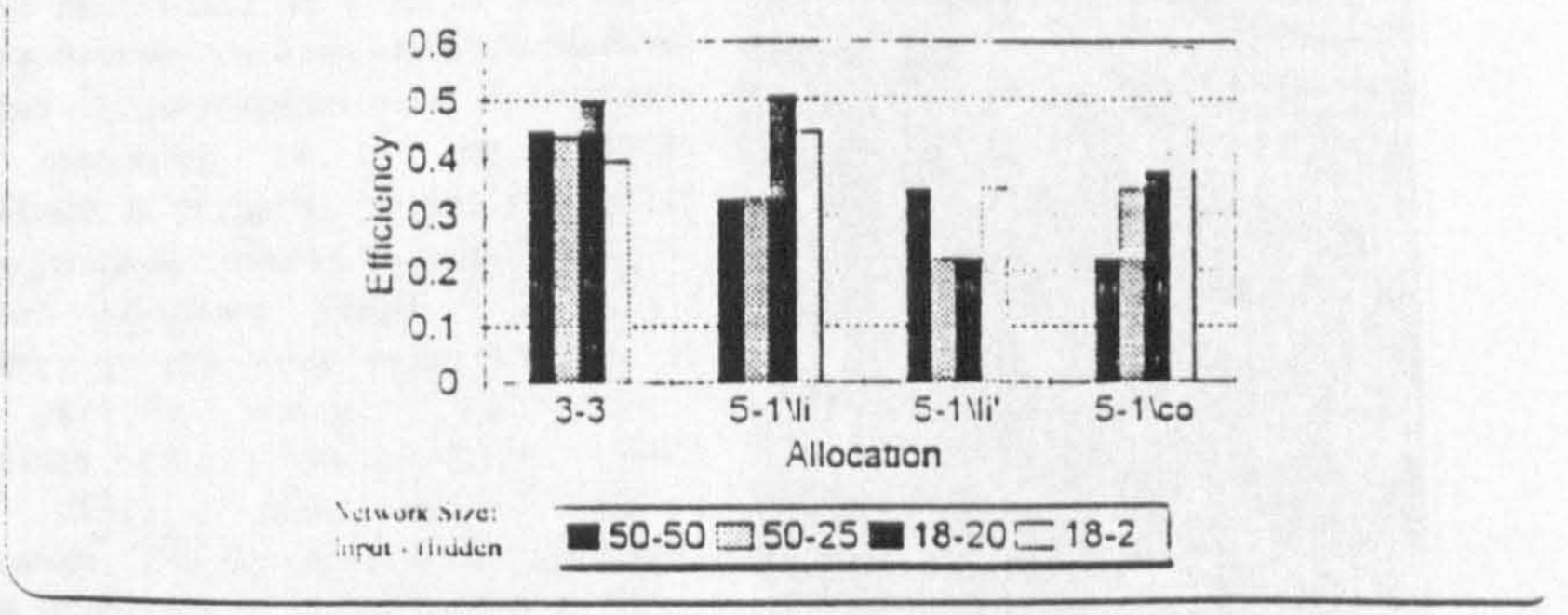

Figure 10 - Efficiency obtained for different transputer structures

expand. Effectively, considering the fact that a neural net requires several hundred iterations to converge, the handling of such large amounts of data is a major problem for an efficient implementation of the learning phase.

5 - Implementation

The system has been implemented on a transputer network consisting of T414 and T800 transputers, as shown in Figure 8. Images are fed to the network by an external program, running in parallel on a host computer, which can access a Matrox frame grabber board. Edge detection algorithms are implemented using a data parallelism structure, splitting the image over n processors (Figure 8). This is done according to the tree structure shown. The mapping of the neural network on the transputer array is done in layers and implemented as a pipeline. This results in overlapping process computing times. Each process includes a layer, so increasing the number of layers in the neural network is done by simply adding intermediate processes. The structure used allows a pipeline implementation of the network using the internal ring (R 1.0 2.0), or the external ring (R 1.0 1.1 ... 2.0). For single layers it allows the data parallelism between the transputers 1.0 and 2.0. It also allows each layer to be mapped on a set of transputer, dividing between them the coupling matrices, to speed up computation time. Experiments were conducted, as to the different possibilities of mapping, the seven processes onto a transputer network ranging from 1 to 7 allocated transputers. Figure 9 shows the increment in efficiency (ratio between the times used to compute x lines with n transputers and the time taken by a single one). This is shown for 1,2,4,and 7 transputers. The use of 5 or 6 transputers causes imbalance in the network and thus is not implemented. This has been done for both layers, and referring to the pipeline configuration, using 3 transputers for the first

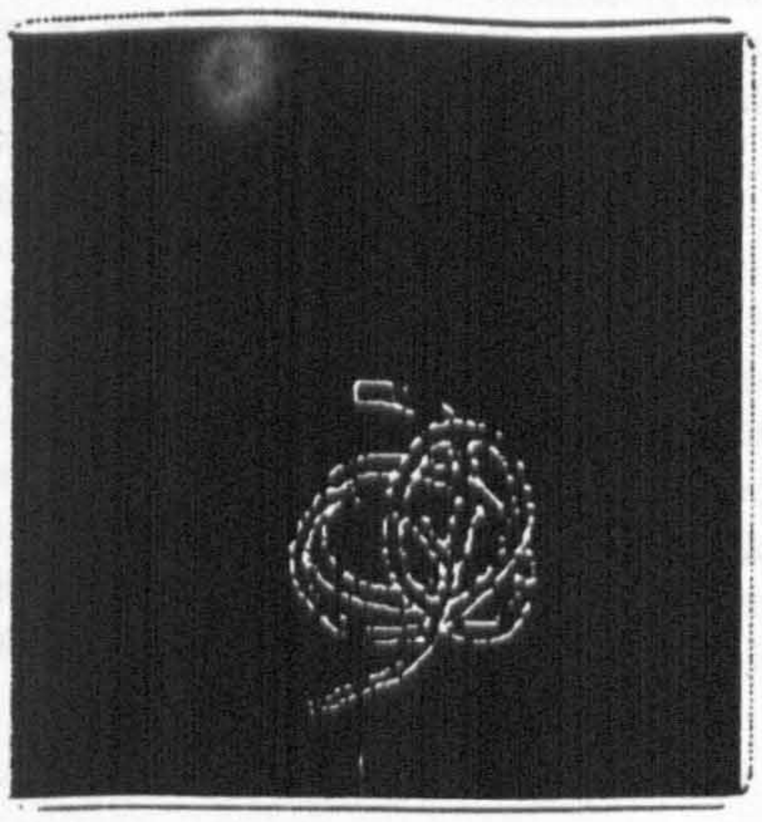

Figure 11 - Arbitration Example

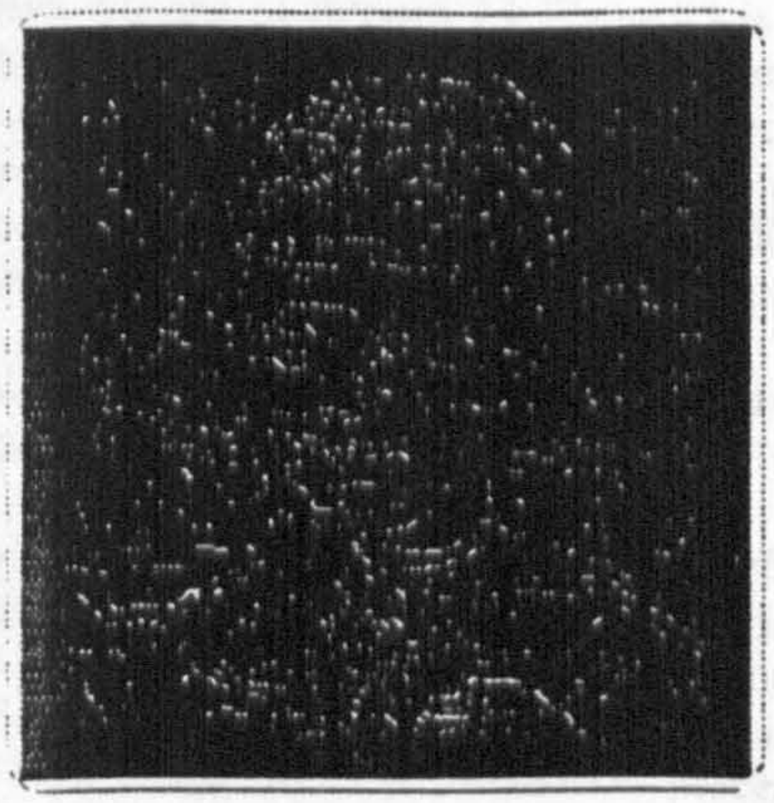

Figure 12 - Arbitration example (I)

Figure 13 - Arbitration example (II)

and second layers. It is clearly visible that the increment in computing speed. is not proportional to the resources used. It is also clear that, increasing the number of transputers from 4 to 7 results in no increase of performance. This is due to the smaller size of the second layer used in all networks, since the output of all networks, is 1 node. This clearly shows that a different allocation of resources is needed. Figure 10 shows the efficiency obtained when more than one transputer is used per layer and process allocation is varied. The first case allows the results for 3 transputers per layer. The other results relate to 5 and 1 transputer per layer performing different amounts of line and column multiplications. The structures tested, were based on the pipeline using the internal ring. They differ by the splitting of the different layers by the nearest transputers (according to figure 8). The others configurations are the reinforcement of the resources allocated to the first layer, were all four transputers are allocated. Two different nodes allocation, when the matrices were split by lines between the transputers are shown (li and li') as the splitting by columns.

6 -Results

Examples of arbitrated edges are presented in figure 11 through 13. Figure 11 is the edge map arbitrated from figures 3 and 4. This network was trained with the set composed by figures 3,4 and 2, and, represent the success rate of the learning set. Learning was performed as a raster scan, with an incremental step of two in both directions. It was performed the removal of background points. The success rate achieved was 1/385 points. The application of this network to arbitrate figures 5 and 7, gives the result present in figure 12.

A more complete set of learning points was used in the learning phase of the network used in the arbitration for figure 13. This consist of 5546 points. From this set 91% recalling rate was achieved by the network.

The complexity of the images used, will make a quantitative comparison difficult to achieve, beeing traditional criteria [6] impossible to apply. By the comparison of the images presented it is clearly visible that most part of the background artefacts, present in Canny's edge map (figure 7) due to the noise, have been removed. Comparing with Robert s operator, a thinner edge line is obtained from the arbitration process. This compares favourably with the technique of applying thresholding after Roberts operator so as to obtain a binary image results in thicker edges.

A careful comparison between figures 3,4, and 9 reveals that some gaps have partially been filled in by the arbitration process. Some of the descontinuities are still present as the area covered by the arbitration network (3x3 window) is small.

7 - Conclusion

We have shown that, distinct edge detection algorithms originate different yet distinct edge maps. These maps present some details that are particular to the output of each unique method. Also, the artefacts introduced by noise have diverse localisation effects due to the different frequency response of the methods. This suggests the fact that distinct algorithm responses could be integrated by an appropriate arbitration system. Although the complexity of the system is augmented, parallelization of the algorithms and an accurate division over an appropriate number of processors will minimise this problem. Our investigations into the use of a neural network, as an edge arbitrator, suggests that it is the ideal solution to the arbitration problem. Although some information could be lost in the

process. results presented. show that some improvement can be achieved using this approach. It is our conviction that some of the imperfections presented by the arbitrated edge maps could be avoided by the use of windows covering a bigger area.

8 -Acknowledgements

MASN Ramalho would like to acknowledge the grant BD-1719 from Programa Ciência. JNICT. Portugal. under which his part of the work was carried out.

9 -References

[1] Canny. J - A Computational Approach to Edge Detection. J Canny. IEEE PAMI-8. n°6. Nov 1986

[2] Deriche. R - Opumal Edge Detecuon Using Recursive Filtering; 1st Internauonal Conference on Computer Vision. London England. pp 501-505. 1987

[3] Deriche. R - Fast algorithms for low-level vision. IEEE PAMI n° 12. vol 1. Jan 1990

[4] Marr. D; Hildreth. E - Theory of Edge Detection. Proc Royal Soc of London. ser B. n° 207. pp 187-217. 1980

[5] O'Gormann. F - Edge Detection Using Walsh Functions. Artificial intelligence. n° 10.pp 215-223. 1978

[6] Pratt. WK: Digital Image Processing. 2nd Edition: John Wiley and Sons 1991

[7] Prewitt. JMS: *Object Enhancement and Extraction in Picture Processing and Psychopictorics. Eds* BS Lipkin and A Rosenfeld. Academic press. pp 75-151. 1970

[8] Roberts. L - Machine Perception of Three Dimensional Solids. 1965

[9] Zurada. JM - Introduction to Artificial Neural Systems. West Publishing Company. 1992

EDGE DETECTION USING NEURAL NETWORKS: A COMPARISON

MASN Ramalho and KM Curtis

Parallel Processing Specialist Group
Electrical and Electronic Engineering Department
University of Nottingham, UK
@mail : mar@eee.nott.ac.uk

ABSTRACT

Different edge detectors present distinct and different responses to the same image thus showing different detail. We are researching the hypothesis of using different edge detection techniques, in parallel, for generalised edge detection of different image types. This allows the attainment of several edge maps, each edge map associated to a different edge detection tecnique containing different features. From the various edge maps edges are arbitrated using a neural network. In this paper, we present examples that show the performance of a neural network edge detector along with other individual edge detection techniques alone. This is done in order to show the advantages of an arbitration approach over the neural network alone and the other techniques.

To overcome the increased computing time, implementation is being performed on a multi-transputer array, using the inherent parallelism of the techniques involved.

1 INTRODUCTION

Many varied edge detection techniques exist. Edge detection is an important problem in image processing applications as it is the common starting step for segmentation or feature detection in images. Each of these techniques have their advantages and disadvantages in the types of scenes which they are capable of analysing. The idea of an edge is difficult to define precisely. Edges having various profiles are usually present in an image. The edge profile usually appears distorted due to the image acquisition device and the noise that always appears in signals. Although in a natural scene the most common borders are vertical and horizontal lines, other shapes are also present and could be more significant in some applications. Many edge detection techniques have been developed to cope with this abundance of different types of edges.

There are some desirable characteristics the an edge detector should achieve such as precision, uniqueness, and robustness to image blurring and noise. In addition, the amount of operations involved should be as small as possible. Some of these characteristics are conflicting. Unfortunately, most of the edge maps which are produced by edge detection algorithms do not fulfil these characteristics. Thus it is common to use some form of post-processing technique to perform single or isolated point removal, gap filling, etc.

Although artificial pictures do not present the difficulties that are present in natural scenes, the comparison of the methods is initially carried out with generated pictures that try to reflect some of the features that are present in digitised video images, such as small blurring, for instance. This is done as we need to know the precise

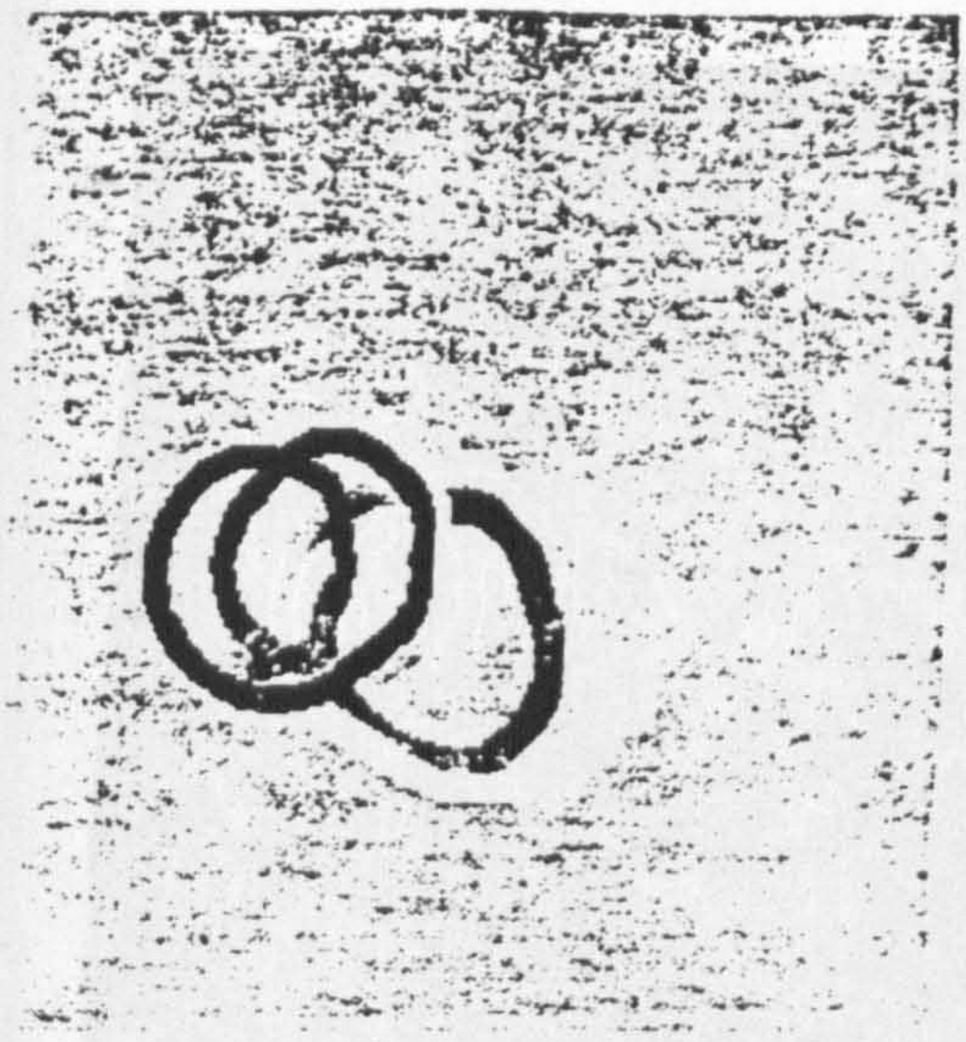

Figure 1: Original Stripe

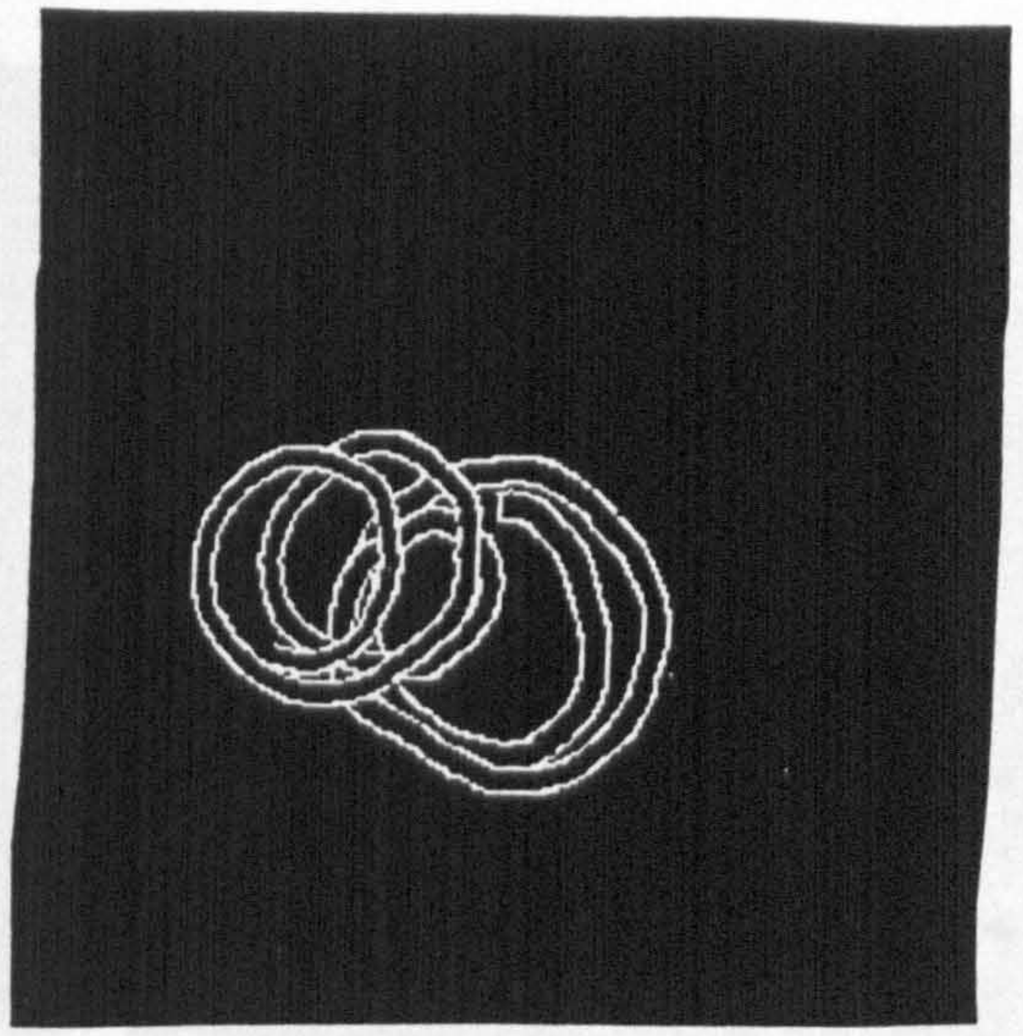

Figure 2: Reference Edges for Figure 1

location of the edges, and thus allows for precise comparisons between the different techniques. An example of such a picture is shown in the first figure presented, Figure 1, which consist of a random radius stripe, where the grey level changes with a random slope. To this picture noise or blurring is added. In the presented case, the picture was blurred with a gaussian function, and a small amount of noise added. Several images were produced and tested. A second image is produced, figure 2, which shows the edge positions in the image.

Our proposal is that different edge detection techniques be utilised in parallel, for generalised edge detection of different image types. This allows the attainment of several edge maps containing different features. From the various edge maps, edges are arbitrated with a neural network performing the merging of the different maps. To overcome the increased computing time, implementation is being performed on a multi-transputer array using the inherent parallelism of the techniques involved.

2 EDGE DETECTION

In the first instance results from a representative number of the various edge detection schemes were implemented in the form of processed images for different types of edge detectors.

Widely referred to methods were selected, as a representative sample[1][2][3], providing a range of approaches. These include derivative approaches, such as Roberts[4], Sobel[1][2] or Prewitt[5] operators. Among other approaches, several optimal filters were implemented such as Marr[6], Canny[7], and Deriche[8].

Figures 3 through 7 show edge maps obtained using two of the above mentioned methods. These maps present some details that are particular to the output of each unique method. The pictures are presented, without any further processing, such as thresholding for instance. Another difference is readily perceived. Although optimal edge detectors mark well defined and closed contours, they will also mark as edges noise related features or weak contrasts due to shadows, for instance, derivative approaches will

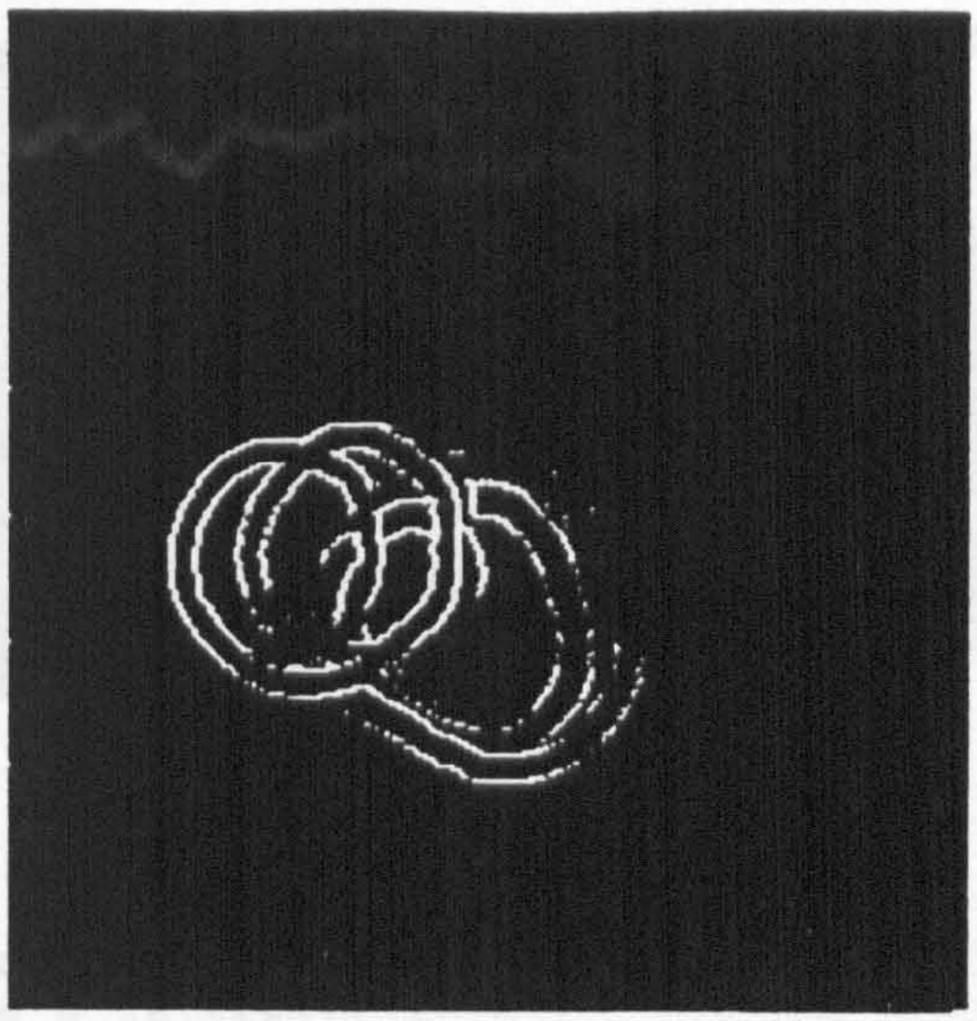

Figure 3: Roberts operator

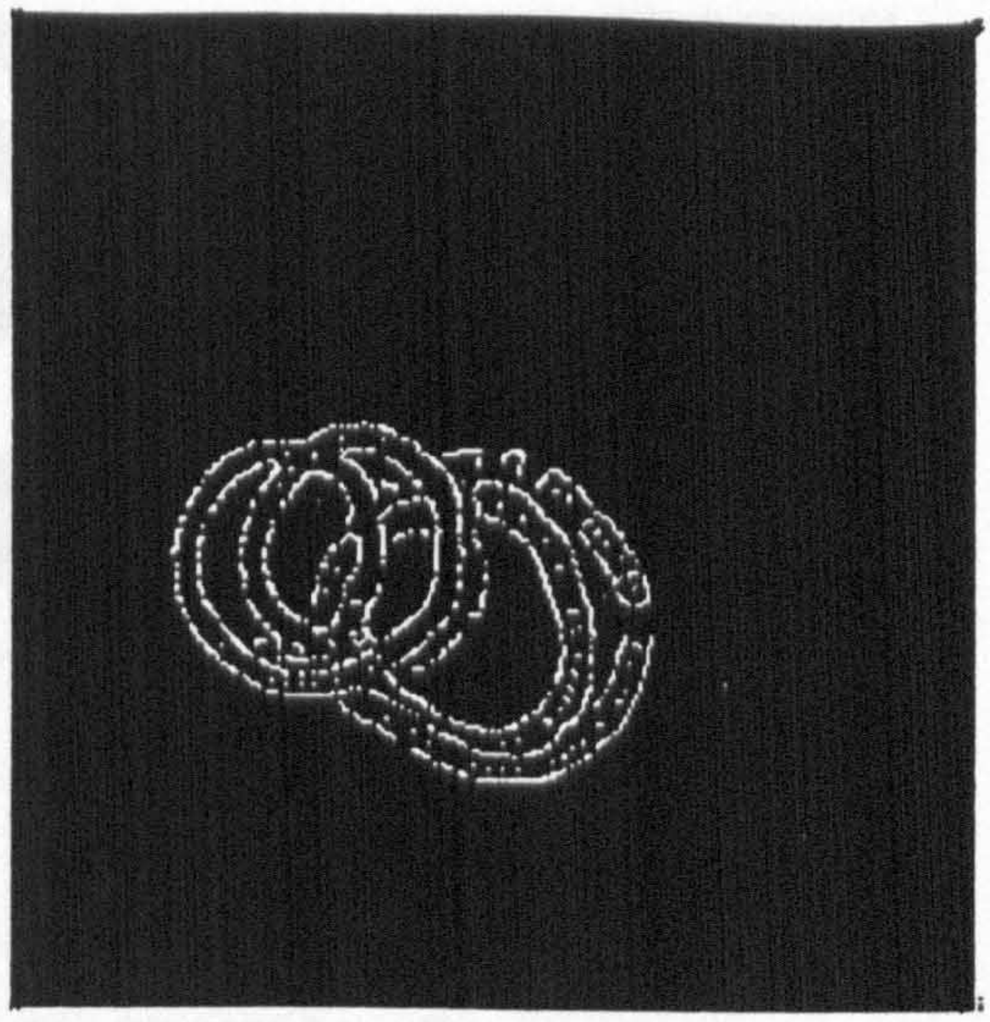

Figure 4: Canny Operator

provide a more effective selection, but at the cost of loosing some weak edges. Also, due to the inherent filtering present in zero crossing operators, derivative approaches have a smaller tendency to dislocate edges. Thus, it can clearly be seen that different methods detect different edge segments. For instance artefacts due to the noise, although partially avoidable by a convenient threshold criteria, are dissimilar in the different maps. This suggests the fact that distinct algorithm responses could be integrated by an appropriate arbitration system in order to fuse the information present in the different edge maps. Several approaches could be used in implementing this process, and, among them artificial neural networks seem to be particularly suited to this job.

Effectively, artificial neural networks can handle incomplete or corrupted sets of data thus they can be applied to the recognition images and can be used to infer the position of missing edges or misplaced edges, based on the knowledge applied by the different edge detection techniques described. To this purpose multi-layer back-propagation neural networks[9] are being used. These types of network are capable of reproducing an input/output relation, learned from a repetitive exposition to a set of examples. They also have an inherently parallel structure allowing for a parallel implementation[10]. Various sizes of neural network architecture's have being investigated for their ability to perform the arbitration required.

3 EXPERIMENTS

The experiments were performed in parallel under the same image conditions the same as possible, in order to allow direct comparisons.

In the neural edge detection case, from the master image a collection of patterns is collected, corresponding to edge and non edge patterns, according to the reference edge map. This set is collected from the images using a raster scan process. A window size is defined and the image is scanned at fixed step.Thus a collection of patterns is produced containing edge patterns and no edge patterns. Vectors which contain only the background windows are discarded. The patterns selected are presented as the training pattern of the neural edge detection system. The network used in te neural edge detection system is a multi-layer neural network. This

Figure 5: Roberts Operator

Figure 6 : girl

Figure 7: Canny Operator

neural network is then trained from that selection of patterns extracted from a reference pair of images. Then, to assess its performance several images, of the same kind, or others, are presented and compared to the edge maps obtained by other edge detectors. It is expected that in the later case, the network will be capable of selecting relevant edges based on the patterns previously learnt.

To perform the arbitration strategy, a couple of algorithms were selected. In the first case, the neural network arbitrate between the Roberts and Canny Operators. First the master image is processed by the selected algorithms. From this image, a collection of patterns are collected, corresponding to edge and non edge patterns, according to a reference edge map. If artificial images are used, the edge map is generated at the same time as the 'master' image, otherwise, it is traced from the original image by the user. This set is collected from the image by the raster scan process previously described. The patterns selected are presented for the training of the arbitration system. The arbitration is done by a multi-layer neural network. This neural network is then trained from a selection of patterns extracted from a reference pair of images. Then, to assess its performance several images, of the same kind, or others, are presented and compared to the edge maps obtained by other edge detectors. It is expected that in the later case, the network will be capable of selecting relevant edges based on the patterns previously learnt.

4 RESULTS

In our experiments a similar image to that shown in figure 1 has been generated simultaneously with a reference edge map (similar to that presented in figure 2). This image was processed by the Roberts and Canny operators. From the original image, the stripe was located and extracted as a set of vectors corresponding to a 3x3 window, which was used as data for the learning phase of a neural network with 9 input points. On the processed images, the same technique is used. The network used now has 18 input points. In the learning phase, the information of the reference window is used to identify the edge or non edge vectors. The network used is a three layer back-propagation neural network, for which several sizes of the hidden layer have been tried.

The networks obtained were then used to process a a number of images for both conditions (direct edge detection, and arbitration). Some of the Results are presented in figure 8 and 9 (edge detection), and 10 through 12 (arbitration).

5 CONCLUSION

We have investigated the ability of a neural network for edge detection arbitration from several edge detection techniques. Problems are substantial for the arbitration system, since it requires that the learning set be extracted from several images, and thus the number of points will be very large. However, the arbitration system has proved to be more efficient than a neural network alone. However it is more computationally expensive. Effectively, considering the fact that a neural net requires several hundred iterations to converge, the handling of such large amounts of data is a major problem for an efficient implementation of the learning phase. Finally results have been presented as to the advantages of an arbitration strategy in comparison with a neural network alone.

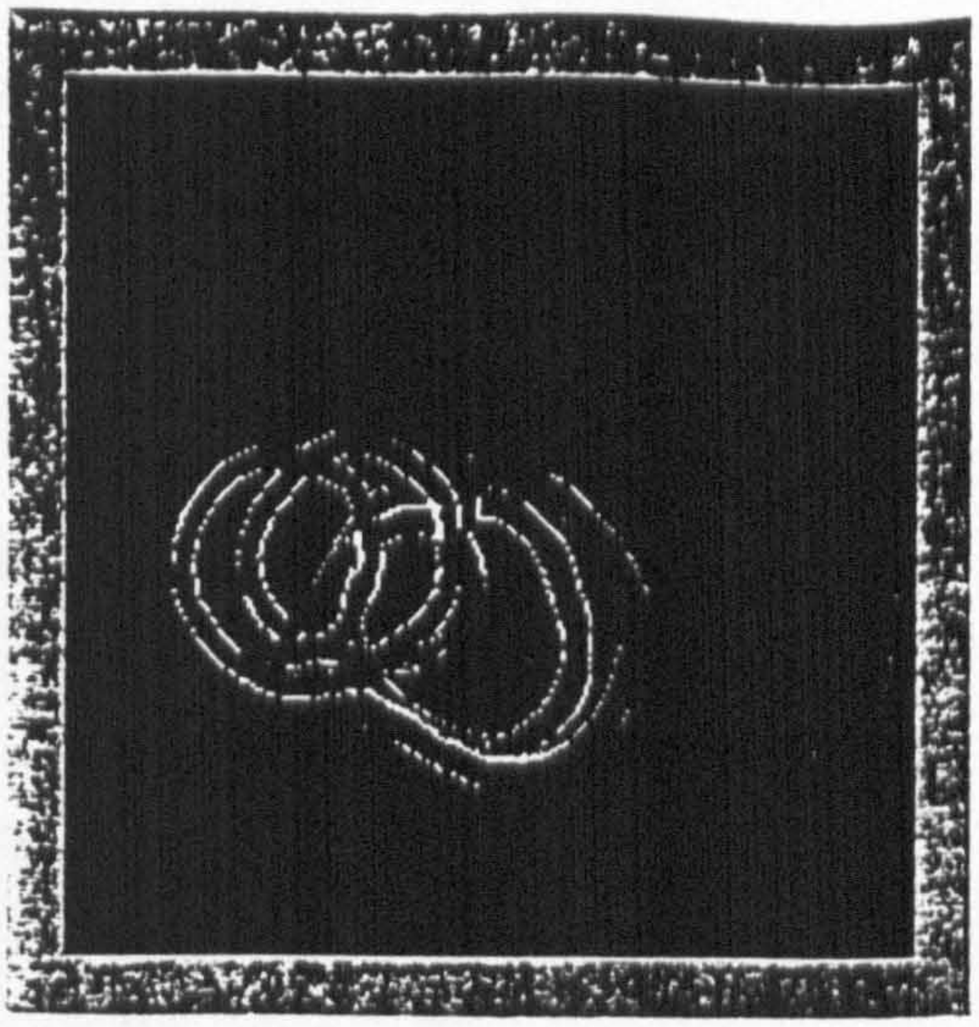

Figure 8: Edge Detection for random stripe(fig 1)

6 ACKNOWLEDGEMENTS

MASN Ramalho would like to acknowledge the Grant BD-1719 from Programa Ciência, JNICT, Portugal, under which his part of the work was carried out.

7 REFERENCES

1. Gonzalez, R and Wintz, P, 1987
"Digital Image Processing", 2nd Edition, Addison-Wesley Publishing Company, USA

2. Pratt,W, 1991
"Digital Image Processing", 2nd Edition, John Wiley and Sons, INC , USA

3. Sonka, M, Hlavac, V and Boyle, R, 1993
"Image Processing Analysis and Machine Vision", Chapman & Hall Computing, UK

4. Roberts, L 1965
"Machine Perception of Three Dimensional Solids", in Optical and Electro-Optical Information Processing, Tippet, JT (Ed), MIT press, USA

5. Prewitt, J, 1970
"Object Enhancement and Extraction in Picture Processing and Psychopictorics" in picture Processing and Psychopictorics, Eds BS Lapkin and A Rosenfeld, Academic Press

6. Marr, D and Hildretn E, 1980
"Theory of Edge Detection", Proc Roy Soc Lon, B-207, 187-217

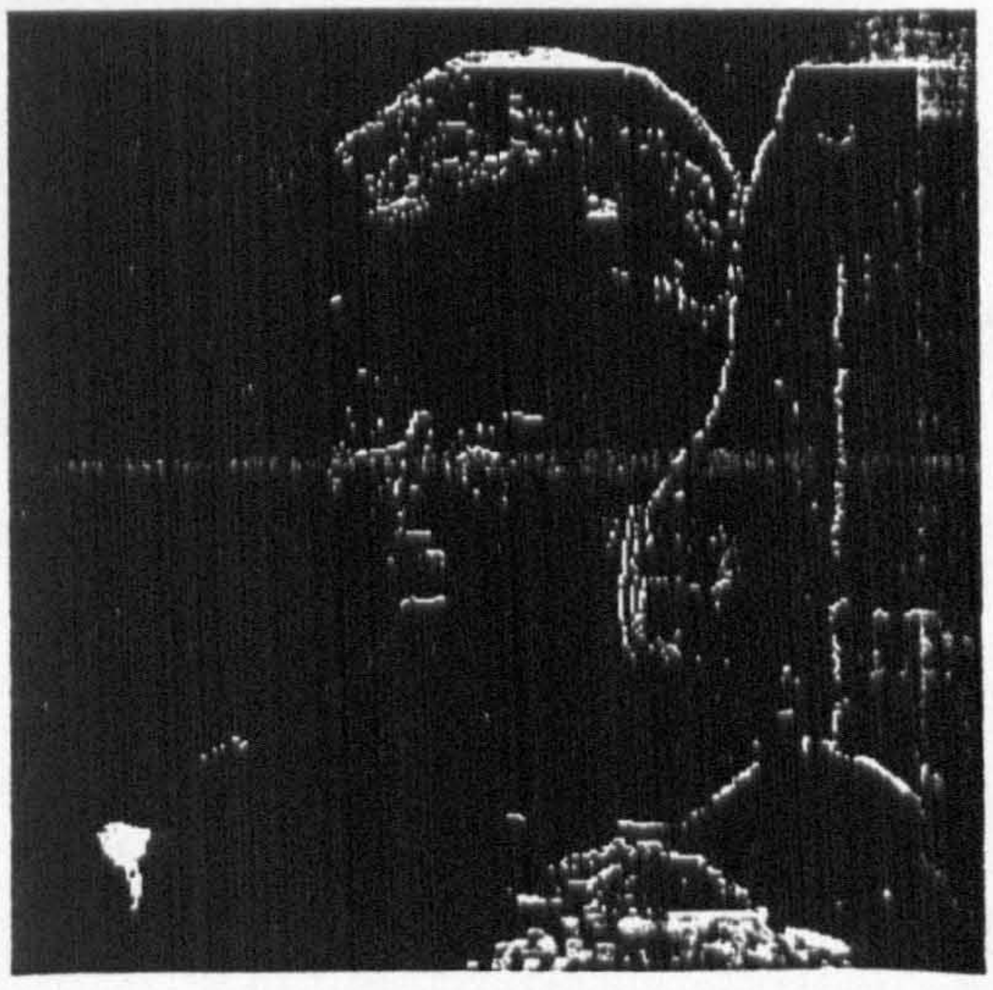

Figure 9: Edge Detection for Girl

7. Canny, J, 1986
"A Computational Approch to Edge Detection",
IEEE PAMI, 8-6,

8. Deriche, R, 1987
"Optimal Edge Detection Using Recursive
Filtering", 1st ICCV, London, UK

9. Zurada, JM, 1992
"Introduction to Artificial Neural systems",
West Publisihing Company

10. Ramalho, M and Curtis, KM, 1994
"Neural Network Arbitration of Edge Maps", in
Transputer Applications and Systems'94;
Gloria, A, Jane, M and Marini, D. (Eds), IOS
Press, Netherlands

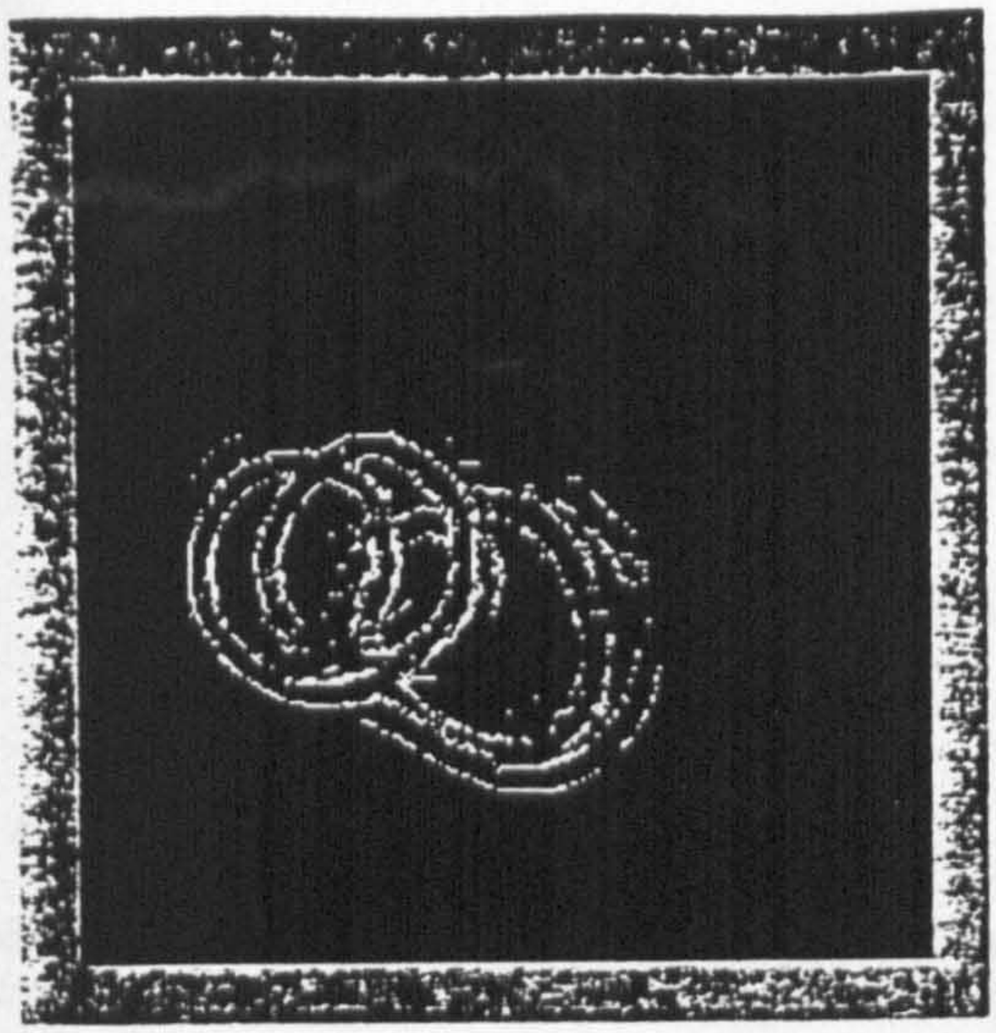

Figure 10: arbitration example
from random stripe (Figure 1)

Figure 11: Arbitration example
(Girl)

Figure 12: Arbitration Example
(II) - (Girl)

EDGE DETECTION USING NEURAL NETWORK ARBITRATION

MASN Ramalho and KM Curtis

Parallel Processing Specialist Group, Elect.Elect.Eng.Dep, University of Nottingham, UK

ABSTRACT

Edge detection is a common starting step in many image processing applications. Many varied edge detection techniques have been proposed. Different edge detectors present distinct and different responses to the same image, showing different detail. Our proposal is that different edge detection techniques be utilised in parallel, allowing for the attainment of several edge maps containing different features. The merging of the different maps is performed by a neural network arbitrator. To overcome the increased computing time, implementation is being performed on a multi-transputer array, using the inherent parallelism of the techniques involved. In this paper, we present examples that show the performance of the neural arbitration of different techniques, along with a comparison with the individual techniques alone.

1 INTRODUCTION

Edge detection is a common starting step in many image processing applications. Effectively, many segmentation and feature detection approaches rely on an edge map of an image. Although a common research topic, the complexities that arise from natural scenes, still leaves the problem unsolved. Indeed, the idea of an edge is difficult to define precisely as they do not correspond entirely to an image feature. Although usually assumed as a discontinuity in the grey level, due to blurring or noise that are inherent to the image acquisition devices available, it is difficult to obtain clear local discontinuities in the grey level as it is impossible to define a standard profile that characterises an edge. This abundance of profiles and imaging conditions makes their detection incomplete or inaccurate in many common situations. Many edge detection techniques have been developed to cope with this abundance of different profiles of edges.

Different types of scenes and imaging requirements will make some of the techniques more suitable for some applications. This is due, not only to the underlying image model assumed, but also to the diverse robustness and sensitivity of each of the techniques. Effectively, depending on a variable number of factors several conflicting requirements are searched for in an edge detection algorithm, these could be, for instance, robustness, resolution, precision or uniqueness. Uniqueness and resolution are important factors that will alleviate the burden imposed on subsequent parts of an image processing system. The marking of closed contours will diminish the uncertainty areas in a segmentation system. Precision is a fundamental requirement for quality control applications, for instance. Also, the complexity and size of an image require that algorithms preferably involving a small number of simple operations, are used to allow large throughputs of the system. Unfortunately, most of the edge maps that are produced by edge detection algorithms do not fulfil these characteristics. To achieve these requirements it is common to use some form of post-processing technique to perform single or isolated point removal, gap filling, etc. This collection of conflicting requirements, makes the edge detection comparison a difficult task to achieve, as the multiplicity of situations that arise in natural scenes makes their comparison difficult. Although some particular approaches have clearly strong points for some particular feature, they have weak points for others. This comparison could be made on a feature by feature basis, but the wide list of situations will make any weighted figure of merit meaningless in a wider context.

2 EDGE DETECTION

Firstly a representative number of the various edge detection schemes were implemented in the form of processed images for different types of edge detectors. Widely referred to methods were selected [1][2][3], providing a wide range of approaches. Derivative approaches, namely the Roberts[4], Sobel[1][2] and Prew itt[5] operators were applied, as were the Marr[6], Canny[7], and Deriche[3] optimal filters. This selection was made on the basis of the computational simplicity of the derivative approaches, and the uniqueness of the

Image Processing And Its Applications, 4–6 July 1995
Conference Publication No. 410, © IEE 1995.

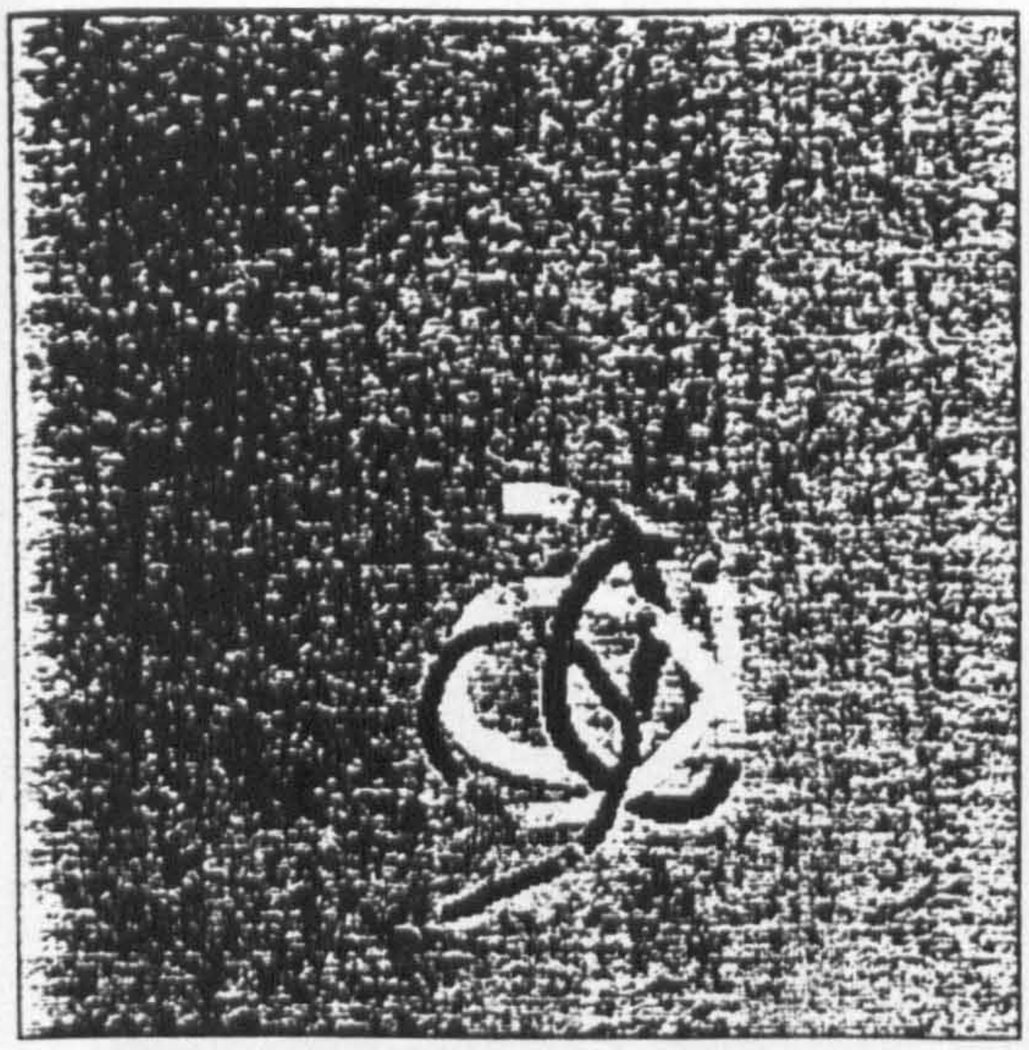

Figure 1: Original Stripe

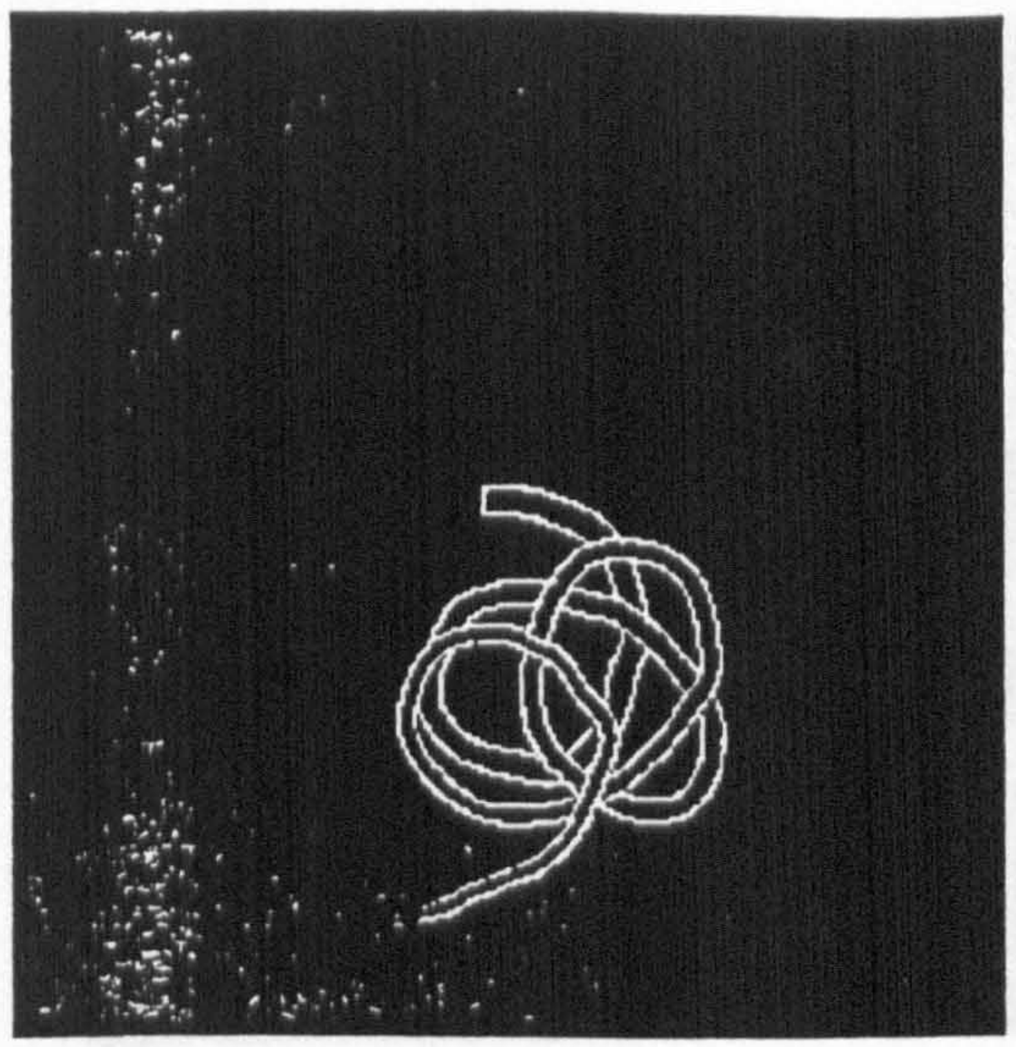

Figure 2: Reference Edges for Figure 1

marked edges produced by the zero crossing of second derivative. Figures 5 through 8 show edge maps obtained using two of the above mentioned methods. The pictures are presented, without any further processing. These maps present some details that are particular to the output of each unique method. For instance, artefacts due to the noise, although partially avoidable by a convenient threshold criteria, are dissimilar in the different maps. Another striking difference is also readily perceived. Although optimal edge detectors mark well defined and closed contours, they will also mark as edge's noise related features or weak contrasts due to shadows, for instance, derivative approaches will provide a more effective selection, but with the cost of loosing some weak edges. Also, due to the inherent filtering present in zero crossing operators, derivative approaches have a smaller tendency to dislocate edges. Thus, it can clearly be seen that different methods detect different edge segments. This suggests that distinct algorithm responses could be integrated by an appropriate arbitration system to fuse the information present in the different edge maps.

The adverse characteristics present in images are, in many applications, common to the type of images used, not only because some of the characteristics were produced by the particular type of devices used, but also due to the fact that some characteristics are inherent to the type of

Figure 3: 'Peppers' image (original)

Figure 4: 'Lenna' image (original)

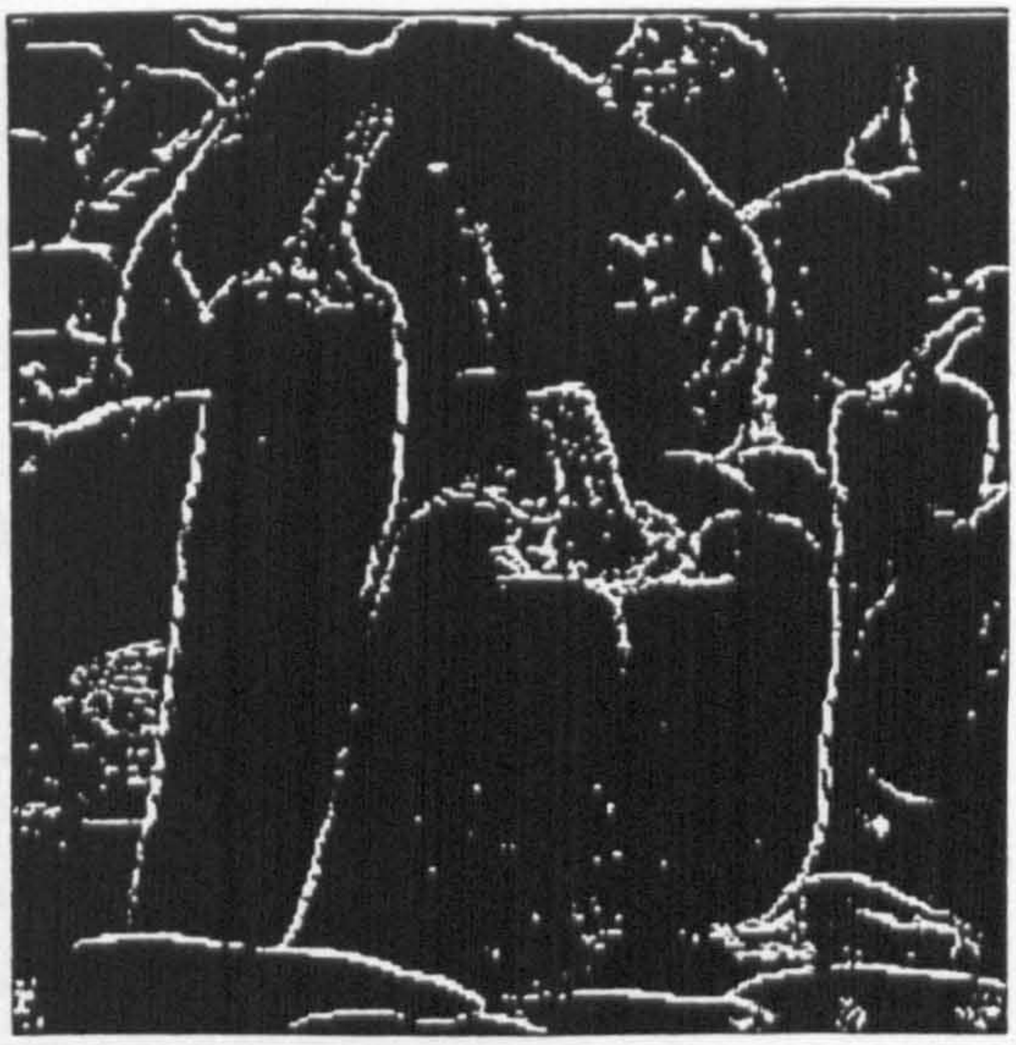

Figure 5: 'Peppers' using Roberts Operator

Figure 6: 'Peppers' using Canny Operator

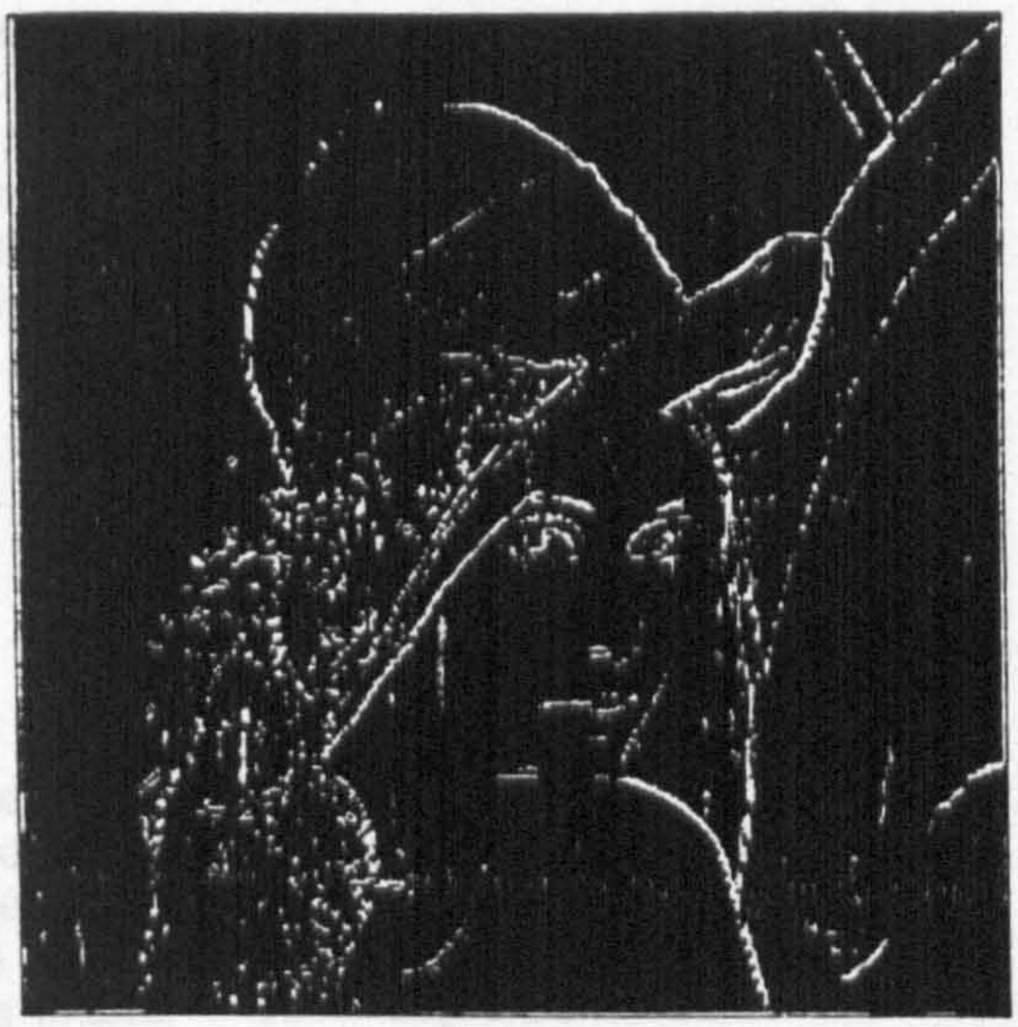

Figure 7: 'Lenna' using Roberts Operator

Figure 8: 'Lenna' using Canny operator

scene being analysed. This will allow us to develop algorithms that can be adapted or fine tuned for particular imaging conditions, assumed constant in a particular application. Also, the merging of the desirable characteristics produced by some operators, even for particular situations could be a way of overcoming the referred problems. Our proposal is that different edge detection techniques be utilised in parallel, for generalised edge detection of different image types. This allows the attainment of several edge maps containing different features. From the various edge maps, edges are arbitrated with a neural network, performing the merging of the different maps. To overcome the increased computing time, implementation is being performed on a multi-transputer array, using the inherent parallelism of the techniques involved.

Among the several approaches that could be used in implementing this process, an artificial neural network seems to be particularly suited to this job. Effectively, artificial neural networks can handle incomplete or corrupted sets of data thus they can be applied to the recognition of images and can be used to infer the position of missing edges or misplaced edges based on the knowledge applied by the different edge detection techniques described. To this purpose a multi-layer back-propagation neural networks [9] are being used. These types of networks are capable of reproducing an input/output relation, learned from

Figure 9: Arbitration (Roberts and Canny)

Figure 10: Arbitration (Roberts and Canny)

a repetitive exposition to a set of examples. They also have an inherently parallel structure allowing for a parallel implementation [11]. Various sizes of neural network architecture's are being investigated for their ability to perform the arbitration required.

3 ARBITRATION

The arbitration strategy has been tested with the above referred to algorithms. These tests have been done using artificially generated images, as we need to know the correct position of edges in the image. These images try to reproduce some features present in artificial images, such as noise, blurring, a larger number of contrast edges, and different edge curvatures. An example of such an image is presented in Figure 1, along with the correspondent edge map, generated at the same time as the image itself. Firstly an image is processed by the selected algorithms. The resulting edge maps, were then scanned, producing a collection of patterns, including edge patterns and (the majority) non edge patterns. This was achieved through joining corresponding windows, and matching them with the information present in the respective reference map (similar to Figure 2). These lists are scanned for conflicts, and repeated points discarded. The patterns selected are presented for the training of the arbitration system. The arbitration is done by a multi-layer neural network. Several network sizes are tested, and the most promising selected.

Then, to assess its performance, several images were presented to the neural network arbitrator. These images could be of the same or of a different kind. The result is then compared to the edge maps obtained by other edge detectors.

4 RESULTS

Several sets of vectors were collected from a set of images similar to the ones presented in Figure 1 and 2. These sets correspond to diverse sizes of windows used, namely 3x3, 5X5 and 7x7. Odd sizes are used to avoid imprecision in the edge position, that arise otherwise. This set of vectors was used to teach several three layer back-propagation neural networks, which has a variable size of the hidden layer. The examples presented are for a 7x7 window. These originated a full set of 9240 points, from which the vast majority are non edge points. These were thought until 90% of the initial misclassifications were correct. These networks were then tested on different images, having similar or diverse characteristics from the image used to generate the training set. Results presented are from the later case (Figure 9 and 10). The results were then evaluated, from these results we can see that the neural network arbitration system is effective in improving the achieved edge map.

5 CONCLUSION

We are researching the ability of a neural network for edge detection arbitration from the edge detection techniques investigated. Results were presented as to the effectiveness of a neural network to act as an arbitrator of the edge information so as to correctly specify all the edges in both artificially generated and real images.

Our investigations into the use of a neural network as an edge detector arbitrator suggests that it is the ideal solution to the problem. Problems are substantial for the arbitration system, since it requires that the learning set be extracted from several images, and thus the number of points will be very large. Effectively, considering that a neural net requires several hundred iterations to converge, the handling of such large amounts of data is a major problem for an efficient implementation of the learning phase. However, the arbitration system has proved to be more efficient than a neural network alone [10], Although apparently more computationally expensive, using the edge maps allows a selection of the points were the neural network should be applied, in orede to reduce the amount of processed points. This strategy will make it competitive to the neural network alone. This problem can be further alleviated by the parallel implementation [11] of the system.

6 ACKNOWLEDGEMENTS

MASN Ramalho would like to acknowledge the Grant BD-1719 from Programa Ciência, JNICT, Portugal, under which his part of the work was carried out.

7 REFERENCES

1. Gonzalez, R and Wintz, P, 1987
"Digital Image Processing", 2nd Edition, Addison-Wesley Publishing Company, USA

2. Pratt,W, 1991
"Digital Image Processing", 2nd Edition, John Wiley and Sons, INC , USA

3. Sonka, M, Hlavac, V and Boyle, R, 1993
"Image Processing Analysis and Machine Vision", Chapman & Hall Computing, UK

4. Roberts, L 1965
"Machine Perception of Three Dimensional Solids", in Optical and Electro-Optical Information Processing, Tippet, JT (Ed), MIT press, USA

5. Prewitt, J, 1970
"Object Enhancement and Extraction in Picture Processing and Psychopictorics" in picture Processing and Psychopictorics, Eds BS Lapkin and A Rosenfeld, Academic Press

6. Marr, D and Hildreth E, 1980
"Theory of Edge Detection", Proc Roy Soc Lon, B-207, 187-217

7. Canny, J, 1986
"A Computational Approch to Edge Detection", IEEE PAMI, 8-6,

8. Deriche, R, 1987
"Optimal Edge Detection Using Recursive Filtering", 1st ICCV, London, UK

9. Zurada, JM, 1992
"Introduction to Artificial Neural systems", West Publisihing Company

10. Ramalho, M and Curtis, KM, 1993
"Integrating Parallel edge detection Schemes using Neural Network Arbitration", ICSPAT'93, USA

11. Ramalho, M and Curtis, KM, 1994
"Neural Network Arbitration of Edge Maps", in Transputer Applications and Systems'94; Gloria, A, Jane, M and Marini, D. (Eds), IOS Press, Netherlands